

SCHOOL OF
CIVIL ENGINEERING
INDIANA
DEPARTMENT OF HIGHWAYS

JOINT HIGHWAY RESEARCH PROJECT

JHRP-77-26

COMPUTERIZED SLOPE STABILITY ANALYSIS
FOR INDIANA HIGHWAYS VOL. II: STABL2
PROGRAM

Eva Boutrup



PURDUE UNIVERSITY

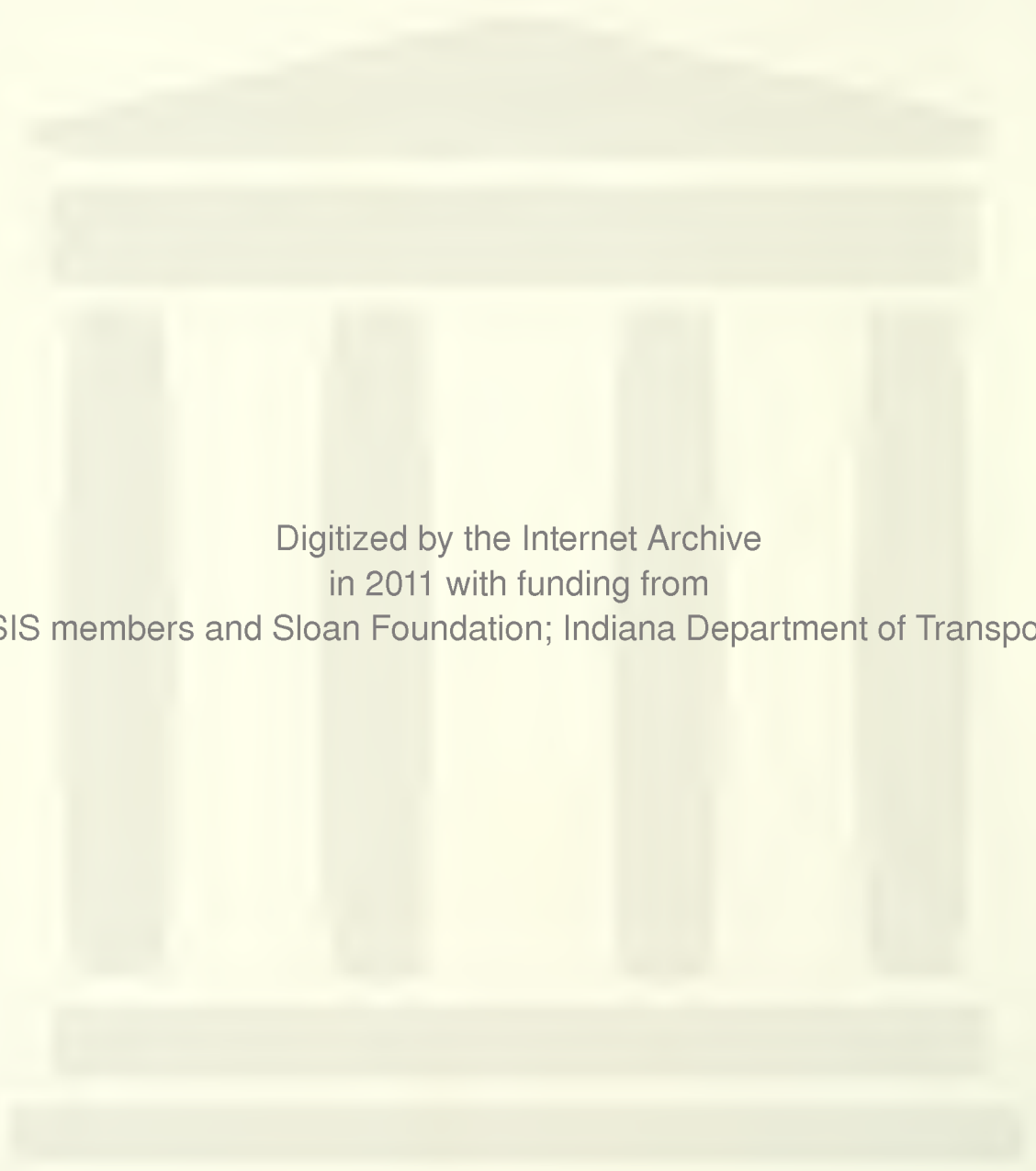


JOINT HIGHWAY RESEARCH PROJECT

JHRP-77-26

COMPUTERIZED SLOPE STABILITY ANALYSIS
FOR INDIANA HIGHWAYS VOL. II: STABL2
PROGRAM

Eva Boutrup



Digitized by the Internet Archive
in 2011 with funding from
LYRASIS members and Sloan Foundation; Indiana Department of Transportation

<http://www.archive.org/details/computerizedslop02bout>

Final Report
COMPUTERIZED SLOPE STABILITY ANALYSIS
FOR INDIANA HIGHWAYS
Vol. II: STABL2 PROGRAM

by
Eva Boutrup
Graduate Instructor in Research

Joint Highway Research Project
Project No.: C-36-36L
File No.: 6-14-12

Prepared as Part of an Investigation
Conducted by
Joint Highway Research Project
Engineering Experiment Station
Purdue University
in cooperation with the
Indiana State Highway Commission

Purdue University
West Lafayette, Indiana
December, 1977

NOTICE

This Volume contains the following Appendices of this Final Report as originally prepared:

<u>Appendix No.</u>	<u>Title</u>	<u>Pages</u>
D	Corrections, Adjustments and Smaller Changes in STABL	373-392
E	Listing of Revised STABL Program (STABL2)	393-507
F	STABL on the IBM Computer	508-512

The following Appendices in the original research report are not contained in these copies of Volumes I and II:

<u>Appendix No.</u>	<u>Title</u>	<u>Pages</u>
A	Tables and Figures Related to Chapter III	211-296
B	Tables and Figures Related to Chapter V	297-319
C	Tables, Figures and Analysis Related to Chapter VI	320-372

A xerox copy of the above Appendices may be obtained for the duplication, mailing and handling costs by inquiry to:

Joint Highway Research Project
Civil Engineering Building
Purdue University
West Lafayette, Indiana 47907

APPENDIX D

CORRECTIONS, ADJUSTMENTS, AND SMALLER CHANGES IN STABL

Besides the direct modifications of the STABL program described in Chapter IV, some corrections, adjustments, and smaller changes have been made in the program during the period from October 1975 to June 1977. A chronologic listing of all changes, including the modifications described in Chapter IV, is found in Table D.1.

D.1 ADJUSTMENTS FOR THE IBM COMPUTER

Some of the changes are related to the conversion of STABL from the CDC 6500 computer at Purdue University to the IBM 360 computer at ISHC's Computer Center, Indianapolis. This is the case for changes number 5 and 6 in Table D.1. It was felt that whenever a change to adjust to the IBM computer was also compatible with the CDC computer, it should be used in both program versions. This reduces the differences between the two program versions.

D.2 DEBUGGING OF STABL

D.2.1 Errors in the Anisotropic Soil Option

Direct errors were found in the anisotropic soil option, where in subroutine ANISO the check of increasing order of anisotropic direction ranges stopped after the second range due to an incorrect "GO TO" statement number. Also, the statement to print out the data was passed over,

again caused by an incorrect "GO TO" statement number. These two errors did not result in actual errors in the stability calculations. However, the direction ranges are specified in degrees, and were later compared to angles in radians. This caused the strength parameters defined for the first anisotropic direction range to always be used, no matter which direction range applied to a particular part of the failure surface. This error was corrected by converting the angles for the direction ranges from degrees to radians before returning from subroutine ANISO. Further, an error in subroutine FACTR had the result that the first anisotropic direction range would not be assigned its anisotropic strength parameters. These two errors had the combined result that the isotropic soil strength parameters were used instead of the anisotropic. This error has been corrected.

In addition, the program is modified so that an anisotropic soil with only one direction range cannot be defined, since this is meaningless. This is reflected in a change in error code AI05.

D.2.2 Infinite Loop in the Circular Surface Generator

An error in the circular surface generation routine was found almost by accident.

The case was as follows. A deep circular search was specified with clockwise and counterclockwise direction limits of -60° and -30° , respectively. Also a minimum elevation was specified.

The first line segment for a circular failure surface is generated at an angle θ with the horizontal, and $\Delta\theta_{\max}$ and $\Delta\theta_{\min}$ are determined as described by Siegel (1975a, p. 81 ff.). Next, $\Delta\theta$ is selected between

$\Delta\theta_{\max}$ and $\Delta\theta_{\min}$. At this point, the angle determined by $\theta + \Delta\theta$ is checked against the vertical. If it exceeds the vertical, an overturning slip circle would result, and this is not allowed.

For every new line segment that is generated, the inclination is checked, and if it exceeds the vertical the following adjustment is made.

First it is determined if it is possible to generate a surface that does not turn over, from this particular first line segment. If it is not, the surface is rejected. If it is, $\Delta\theta_{\max}$ is reduced to the current value of $\Delta\theta$ ($\Delta\theta_1$) and a new $\Delta\theta$ ($\Delta\theta_2$) is selected which will be less than the previous one, see Figure D.1. The surface is generated again, starting with the second line segment.

In our particular problem, this new surface happened to extend below the minimum elevation specified ($y_{\min}$) to which a failure surface is allowed to extend. When this happens, the program determines if it is possible to generate a surface from the first line segment that does not extend below $y_{\min}$. If it is not, the surface is rejected. If it is possible, $\Delta\theta_{\min}$ is increased to the current value of $\Delta\theta$ ($\Delta\theta_2$) and a new $\Delta\theta$ ($\Delta\theta_3$) is selected between $\Delta\theta_2$ and $\Delta\theta_1$.

What occurred in this particular problem was that a surface with this first line segment was impossible to complete without either extending below the $y_{\min}$ depth or turning over (exceeding vertical). This would result in a continuous increase in $\Delta\theta_{\min}$ and decrease in $\Delta\theta_{\max}$ until the two values became almost identical.

At this point two possibilities existed. $\Delta\theta_{\max}$ could decrease so much that it was impossible to generate a surface that did not extend below the minimum elevation $y_{\min}$. In this case, the surface would be

rejected (and counted rejected), and a new surface would be generated with a new first line segment. The other possibility was that $\Delta\theta_{\min}$ would increase so much that it was impossible to generate a surface that did not turn over. This situation, however, was not checked, because the check for possible generation of non-overturning surfaces was based on the initial value of $\Delta\theta_{\min}$. So if a new $\Delta\theta_{\min}$, caused by a conflict with $y_{\min}$, prevents generation of non-overturning circular failure surfaces, the program would continue in an infinite loop, decreasing $\Delta\theta_{\max}$ for each loop, but not below the value of $\Delta\theta_{\min}$. This is apparently what happened in the problem that was analysed.

To avoid this situation, an additional counter was introduced, so that no more than 20 trials are allowed to continue a circular failure surface from the first line segment. If the 20 trials are reached, the surface will be rejected and a new surface generated. If more than 200 surfaces are rejected in the attempt to generate a single trial failure surface, the program is terminated. This number was originally 500, but this was felt to be too large.

D.2.3 Unit Weight of Water

The STABL User Manual [Siegel (1975b)] states that any consistent set of units can be used in the STABL program. However, the unit weight of water was arbitrarily defined as 62.4, implying pcf. If units other than pounds and feet were used, the result would not be correct.

To make STABL able to handle other unit systems, the unit weight of water was made an input variable. If the unit system is pounds and feet, the value 0. (real zero) may be specified, in which case the program will use the default value of 62.4.

D.2.4 The TURKEY Trace Key

During the error traces of the STABL program, Mr. Bellinger from the ISHC Computing Center, Indianapolis, introduced a special trace key subroutine, named TURKEY, for "Trace under referral key".

The TURKEY subroutine made it possible by means of a single data card to control

- (a) which trace keys should be activated, and
- (b) the amount of output for a trace key run.

The use of the TURKEY subroutine is illustrated in Table D.2. The parameters are:

- KLO - Integer number, indicating the lowest key number of the trace keys to be activated.
- KHI - Integer number, indicating the highest key number of the trace keys to be activated.
- KMX - Integer number, sets the limit for the total number of trace key activations.
- Z - Real number, a dummy variable, used in the call for TURKEY when an individual trace key has a lower number of variables to trace, than the 14 variables defined for the TURKEY subroutine.

In Table D.2, KLO and KHI are specified as 20 and 40, respectively. The result is that trace keys number 20 through number 40 will be activated, including trace key number 30 as illustrated in the Table. KMX is chosen as 500, limiting the amount of trace key output to a total of 500 activations.

When the computer output indicated in which of the program subroutines an error occurred, the trace key search was limited to this subroutine and eventually related subroutines.

During some of the error tracings, the TURKEY subroutine as shown in Table D.2, was modified slightly. This was done in cases where the error by the initial trace was found to occur in a failure surface generation process, and after a certain number of surfaces had already been successfully generated. The TURKEY subroutine was modified so that KLO would indicate the number of surfaces to be generated, before a concentrated and more extensive trace key search was activated. Contrary to the original use of TURKEY, this required that only those trace keys that were to be used, should be left in the program. This disadvantage, however, could have been eliminated by introducing a fourth integer variable in the call for TURKEY.

D.3 OTHER ADJUSTMENTS

D.3.1 Modifications in the Irregular Surface Generator (RANDOM)

Siegel (1975a) states that whenever an irregular surface intersects a surface generation boundary (specified with command code LIMITS), the angle limitations for surface generation are modified accordingly, and a new direction for the last line segment is selected. This, however, was not done.

The original STABL program treated this case as follows:

- (1) If a total surface was generated, but an intersection with a limitation boundary occurred for the last line segment, the whole surface was rejected.
- (2) If at any point during the generation of a surface, intersection with a limitation boundary occurred, the following procedure was applied:

- (a) If the boundary was downward deflecting, the surface was rejected.
- (b) If the boundary was upward deflecting, a new direction for the last line segment was tried, but without modifying the initial angle limitations.

After the modifications, STABL treats the case as follows:

- (1) If a total surface is generated, and an intersection with a limitation boundary occurs for the last line segment, another line segment is selected in the direction of the upper angle limitation if the boundary is upward deflecting, and in the direction of the lower angle limitation if the boundary is downward deflecting.
- (2) If at any point during the generation of a surface, intersection with a limitation boundary occurs, the following procedure is used:
 - (a) If intersection occurs with a downward deflecting boundary, the upper angle limitation θ_u , is decreased to the current value of θ , and a new θ is selected between the lower angle limitation θ_L , and θ_u . See Figure D.2. If this new line segment still intersects a limitation boundary, a third line segment is tried with $\theta = \theta_L$ if the boundary is downward deflecting, and $\theta = \theta_u$ if the boundary is upward deflecting.

If this line segment also intersects a limitation boundary, the surface is rejected.

- (b) If the intersection occurs with an upward deflecting boundary, the lower angle limitation θ_L , is decreased to the current value of θ , and a new value of θ is selected between θ_L and θ_u .

If this new line segment also intersects a limitation boundary, the procedure outlined under (a) is followed.

D.3.2 Modifications of Errors RC11 and BK10

To avoid an infinite loop in the program, if for some reason a surface cannot be successfully generated, a counter is included in all surface generation routines. If a certain number of surfaces are rejected in the attempt to generate a single trial failure surface, the errors RC11 (for circular or irregular surfaces) or BK10 (for sliding block surfaces) will be displayed, and the program terminated.

The number of surfaces that could be rejected before the program was aborted, was originally 500. This number was felt to be too large, and was reduced to 200.

Also it was felt that if most of the surfaces were generated when this error occurred, it would be reasonable to have the results printed out for these surfaces. Hence the program was modified, so that if a minimum of ten surfaces are successfully generated, the results of the ten most critical will be displayed, despite the errors RC11 or BK10.

D.3.3 Format Adjustments

Several changes were made to provide a better format for the output of the STABL results.

D.3.3.1 Format Statements

The message about the ten most critical surfaces, as originally placed in the program, would be activated for the CIRCLE and RANDOM search routines, but not for the BLOCK routine. This was corrected.

Also, the factor of safety was displayed with four digits after the decimal point. Since the value is calculated with an accuracy of ± 0.005 , it seemed unreasonable to display more than three digits after the decimal point. This was done.

D.3.3.2 Misleading Factors of Safety

A more significant modification is that related to surfaces where a misleading factor of safety is obtained.

In the original STABL program these surfaces were displayed with coordinate points, but not with a factor of safety, except for specified surfaces. Further, they were still included in the number of surfaces generated, and if their factor of safety was among the ten most critical, they would be listed among these surfaces. This occurred frequently, because the misleading factors of safety often are low (see Chapter IV, Section 4.5.5). Therefore, if misleading surfaces occurred, one would have to compare the surface coordinates of the ten most critical surfaces with those of the misleading surfaces, to avoid using a misleading factor of safety.

In a few cases all ten most critical surfaces were misleading, so that, even though the program did calculate some factors of safety that were not misleading, none of these would be displayed, and the output would contain no valid results.

To avoid this awkward situation, the program was modified, so that when a factor of safety is misleading, the surface coordinates are printed out as before. Then the misleading factor of safety is printed out, whereupon it is assigned a value of 500 to avoid being counted among the ten most critical. Only if less than ten surfaces with non-misleading factors of safety are generated, will a misleading surface be displayed among the ten most critical, but with a factor of safety of 500.

D.3.3.3 Changes of the Calcomp Plot

The format of the Calcomp plot was altered to include print of title and factor of safety.

After the option of several piezometric surfaces was introduced, it was felt desirable to mark these surfaces specifically, to distinguish them from the soil boundaries. This was provided by writing a "W" (for WATER) on the points that define the piezometric surfaces.

In the same manner, limitation boundaries for surface generation are indicated with an "L" (for LIMITS) and surcharges are marked with a "P".

D.3.3.4 Updating of Comment Cards

The STABL program is very well documented in terms of comment statements in the program itself. These comments explain the functions of the main program and each of the subroutines, and define the variables used in each of these subdivisions of the program.

Some of these comment statements had not been updated according to the latest changes made by Siegel during the development of STABL.

Later, changes and corrections made by the author, were not always followed by updating of comment statements.

A thorough updating of the comment statements were made in November 1976. Changes made after this date were followed by corresponding changes in the comment statements.

D.3.3.5 Flow Chart of STABL

Although the STABL program in itself is well documented with more than 4000 comment statements, it was felt that a flow chart showing the individual steps in the program on a large scale level could be useful. This was made, and is shown on Figure D.3.

Table D.1
Chronologic Listing of Corrections and Changes Made in the STABL Program between August 1975 and June 1977

Change no.	Change(s)	Reason(s) / Consequence(s)	Location in present programs	Date
1	STABL Main: Defined IRLK = 0.	IRLK was not defined before it was used.	STABL: 281	Aug. 1975
2	Subroutine RANSLF: Entry BLKSUF: Added cards between old RANS 1448 & 1450, and old RANS 1808 & 1810.	To prevent duplicated points of a sliding block failure surface on the ground surface.	BLKSUF: 812C 292 to 312, and 812C 436 to 494.	Oct. 1975
3	Subroutine RANSLF: Modified sequence and steps in RANDOM surface generation. a) Removed old RANS 1248 to 1262. b) Added new cards instead of old RANS 1354. c) Added new cards instead of old RANS 1384 & 1386.	To make the RANDOM (irregular surface) generation routine more flexible when intersection with limitation boundaries occur. See details in text.	a) between RANS 1250 & 1252 b) RANS 1342 to 1382 c) RANS 1412 to 1468, 1474	Nov. 1975 (Feb. 76)
4	Subroutine PROFILE, RANDOM and PLOTIN. PROFILE: 364(11), 370(14); Added cards between old 408 & 410; old 420(11), 438(11), 440(11), 442(11), 444(11); Added cards between old 444 & 446. RANDOM: 641(A), Added cards between old 1856 & 1858, Added cards between old 1864 & 1866. PLOTIN: Added cards between old 566 & 568.	To provide additional information printed on the Calcomp plot. - provides print of problem description on Calcomp plot. - provides print of the most critical factor of safety on Calcomp plot. - provides print of factor of safety for specified surface on Calcomp plot.	PROFILE: 364, 409 to 418, 424, 438 to 445. RANDOM: 641, 1858 & 1860, 1865 to 1868. PLOTIN: 565 to 567	Nov. 1975
5 (cont. on next page)	Introduced additional variable in several Do Loop's to define the Do Loop counter outside the Do Loop. Changes in the following subroutines: LOADS: 441(A), 450(1), 453(A), 462(1).	To adjust program to the IBM computer. If a Do Loop counter reaches its maximum value, and is then used later in the program, it is not defined on the IBM computer. On the CDC computer it will take the maximum value.	LOADS: Same as old	Dec. 1975

* (1) = Interchanged, (A) = removed, (A) = Added

See Changes in comment cards not included here, but they are indicated in the program listings, Appendix 2.

Table D.1 (cont.)

Change no.	Change(s)	Reason(s) / Consequence(s)	Location in present program	Date
5	RANDOM: 1331(A), 1342(I), 1355(A), 1364(I), 1370(I), 1384(I), 1390(A), 1552(I), 1599(A), 1684(I). RANSUP: 1051(A), 1091(A), 1114(I), 1136(I). SLICES: 353(A), 370(I). WEIGHT: 441(A), 450(I), 451(A), 465(A), 474(I), 475(A), 541(A), 550(I), 675(A), 688(I).	Subroutine RANSUP was renumbered from card # RANS 1050.	RANDOM: Same RANSUP: 1052, 1092, 1118, 1140. SLICES: Same WEIGHT: 478, 448, 440; 504, 516, 518; 632, 6-2; 732, 7-6	Dec. 1975
6	Subroutine FACTR: 366(I)	Equivalence statement modified to adjust to IBM.	FACTR: Same	Feb. 1976
7	Subroutine SLICES: 522(I), 524(I)	To avoid duplicated division points of slices, which are later eliminated anyway. The change is consistent with other similar statements in this subroutine.	SLICES: Same	Feb. 1976
8	Subroutine PROFILE: 616(R), 620(I)	If only one subsurface boundary were present, the statement said: DO 8 1-N,N-1, which is not a valid statement.	PROFILE: Same	Feb. 1976
9	Subroutines ANISO and FACTR. ANISO: 372(I), 418(R), 434(I), 434(I), 470(I), added cards between 472 & 474. FACTR: 414(I), 420(I), 428(A)	To correct several errors related to the anisotropic soil option. A change in Error A105 was also made, see details in text.	ANISO: Same FACTR: Same	Apr. 1976
10	Subroutine RANSUP: 957(A), added cards between 960 & 962.	To overcome infinite Do Loop caused by a conflict between restrictions on minimum elevation to which a failure surface can extend, and overturning slip surface for circular failure surfaces	RANSUP: 957, 960, 961.	Apr. 1976
11	Changed format for factor of safety from F15.4 to F15.3. Subroutine RANDOM: 1854, Subroutine FACTR: 576	The factor of safety is calculated with an accuracy of ± 0.005	RANDOM: Same FACTR: Same	Apr. 1976

Table D.11(cont.)

Change No.	Change(s)	Reason(s) / Consequence(s)	Location in Present Program	Date
12	Subroutine RANDOM: 1432(11), 1450(1).	Correction: SLP is an inclination (tan to angle), not an angle.	BAUNME Same	July 1976
13	Major changes to include BLOCK2 sliding block search, and up to ten different piezometric surfaces. Also included in this revision is the possibility of using unit systems other than pounds and feet, by defining the unit weight of water as an input variable. Possible number of anisotropic soils was increased to ten. The number of trials before abortion, for generation of a single trial failure surface, reduced from 500 to 200. Major changes in subroutines WATER and WEIGHT. Some changes in STABL Main, and subroutines PROFILE, RANDOM, SLICES, SOILUT, SCALER, PLOTIN, and PLTN. Changes in CORROW/BLK cards in STABL Main and in most of the subroutines. BLASUP separated as an independent subroutine; BLOCK2 added as a subroutine; subroutine WEIGHT renumbered.	BLOCK2 was felt desirable as a complement to BLOCK. The possibility of defining more than one water table was suggested by ISMC, and showed to be a useful tool. See details in Chapter IV. The STABL User Manual states that any consistent set of units can be used in the STABL program. However this was not true for the original STABL program, since the unit weight of water was arbitrary defined as 62.4 (pcf). This is now corrected. The changes require additional data cards and changes in some of the existing data cards	STABL Main: 248, 250, 254, 282, 334, 360, 384, 479, 480, 481. PROFILE: 348, 352, 354, 360, 362, 370(R), 380, 954, 958, 962, 987, 1000 to 1002, 1151, 1176 & 1177. WATER: 126, 128, 138, 182, 209 to 221, 222 to 231, 240, 254, 279, 280, 292 & 294, 308, 310 to 315, 330, 348, 352. RANDOM: 418, 422, 645, 1009 to 1102, 1454 to 1459, 1717 & 1718, 1872 BLASUP: 1508 to 1509. BLASUP: 2, 240 to 262, 628 to 636. BLOCK2: whole subroutine added SLICES: 212, 216, 218, 343, 345 to 350, 353, 359 & 360, 364, 370, 373, 466 to 472, 497, 513, 517, 520 to 528, 543 WEIGHT: 402, 406, 408, 418, 424, 436 to 444, 458 to 464, 534. Trace key between SAR & SDO. From 570 to 702 complete change in sequence of cards SOILUT: 2, 92, 94. SCALER: 148, 152, 154, 200 to 203. PLOTIN: 298, 302, 304, 425 to 428, 431 to 436, 441. PLTN: 242, 246, 248, 448 to 452, 457.	Aug 1976

Table D.1(cont.)

Change no.	Change(s)	Reason(s) / Consequence(s)	Location in present program	Date
14	Subroutine RANDOM: Moved print statement about the ten most critical surfaces from between 1056 & 1060 to between old 1826 & 1828	In the original position, the message was only printed for CIRCLE and RANDOM searches, and not for a BLOCK search.	RANDOM: 1078(1), 1822 to 1828, 1832(1)	Sep. 1976
15	Subroutine FACTR: Added cards between 534 & 536, 557(A), 580(1), 602(1), added cards between old 602 & 604.	To prevent misleading surfaces being included in the ten most critical.	FACTR: 536 to 538, 557, 580 & 591, 602 to 604.	Oct. 1976
16	Subroutine PLOTIR: 429(A), added cards between old 438 & 440, 459(A), added cards between old 466 & 468, 497(A), added cards between old 510 & 512, and old 516 & 518, 523(A), added cards between old 620 & 622, 622(1).	To provide print of letters on Calcomp plot to identify piezometric surfaces (U), limitation boundaries (L), and surcharge loads (P). Also provide on center symbol (*) at each point defining the most critical failure surface.	PLOTIR: 429, 438 & 439, 459, 466 & 467, 497, 510 & 511, 516 & 517, 523, 622 to 640.	Nov. 1976
17	Updating of comment cards in most of the subroutines.	Some of the comment cards in the original STABL program had not been updated according to changes made during the development of the program. Later, changes made by the author, were not always followed up by updating of comment cards.	All changes in comment cards (as well as all other cards) are indicated on the listing of the revised STABL program, Appendix E.	Nov. 1976
18	Subroutine PROFIL: PROF1036, changed format statement from 141 to 140	To avoid waste of paper	PROFIL: 1036	Mar. 1977

Table D.1 (cont.)

Change No.	Change(s)	Reason(s) / Consequence(s)	Location in present program	Date
19	STABL Main: subroutines PROFIL, RANDOM and FACTR. STABL Main: 249(A), 250(I), 254(I), cards added between old 280 & 282, 358(I), 360(I), 384(I), cards added between old 400 & 402, and after old 402, 449(A), cards added between old 466 & 468, and after old 468. PROFIL: 426(I) RANDOM: 631(I), 671(A), cards added in front of old 1828, card added between old 1870 & 1872. FACTR: 363(A), cards added between old 466 & 468, and between old 574 & 578.	To provide the option of using the modified Bishop factor of safety in addition to the simplified Janbu factor of safety, for circular shaped failure surfaces. Problems with the simplified Janbu procedure occurred when applied to deep circular failure surfaces for embankments on soft foundation. The factor of safety may be significantly overestimated. See also Chapter IV. Addition to data cards.	STABL Main: 249, 250, 254 & 255, 283 & 284, 358 & 360, 384, 401 & 402, 404, 449, 467 & 468, 470. PROFIL: Same RANDOM: 631, 671, 1828 to 1830, 1873. FACTR: 363, 467 & 468, 577 to 580.	Mar. 1977
20	Subroutines RANDOM, RANSUF, BLKSUF and BLOCK2 RANDOM: 1483(A), cards added between old 1702 & 1704, and after old 1926, and in front of old 1872. RANSUF: 1500(R), 1502(I) BLKSUF: 630(R), 632(I) BLOCK2: 295(A), 592(R), 594(I).	To modify errors RC11 and BK10, so that if a minimum of ten (10) surfaces are successfully generated, the results of the ten most critical will be displayed, despite the errors.	RANDOM: 1483, 1702 & 1703, 1821 to 1823, 1874. RANSUF: 1502 BLKSUF: 632 BLOCK2: 295, 594	Mar. 1977
21	Subroutine BLOCK2: 508(R), 512(I), 515(R), 516(I), 530(R).	Correction of error in generation of active portion of a failure surface.	BLOCK2: Same	June 1977

Note: A last change was made in January, 1978. See Appendix E, pp. 440 and 443.

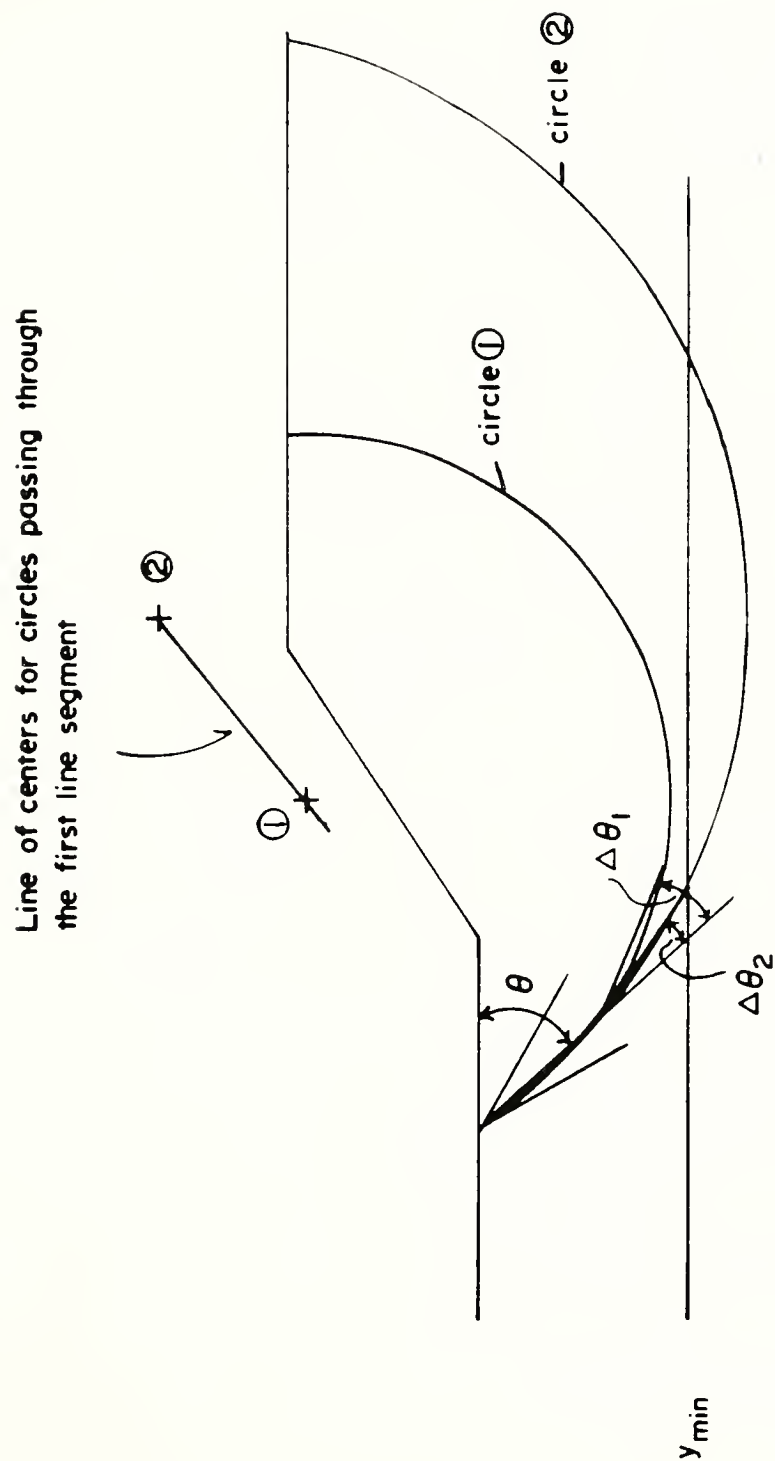


FIGURE D.1 ATTEMPT TO GENERATE A CIRCULAR FAILURE SURFACE WHEN CONFLICT OCCURS BETWEEN OVERTURNING SLIP SURFACE AND MINIMUM ELEVATION

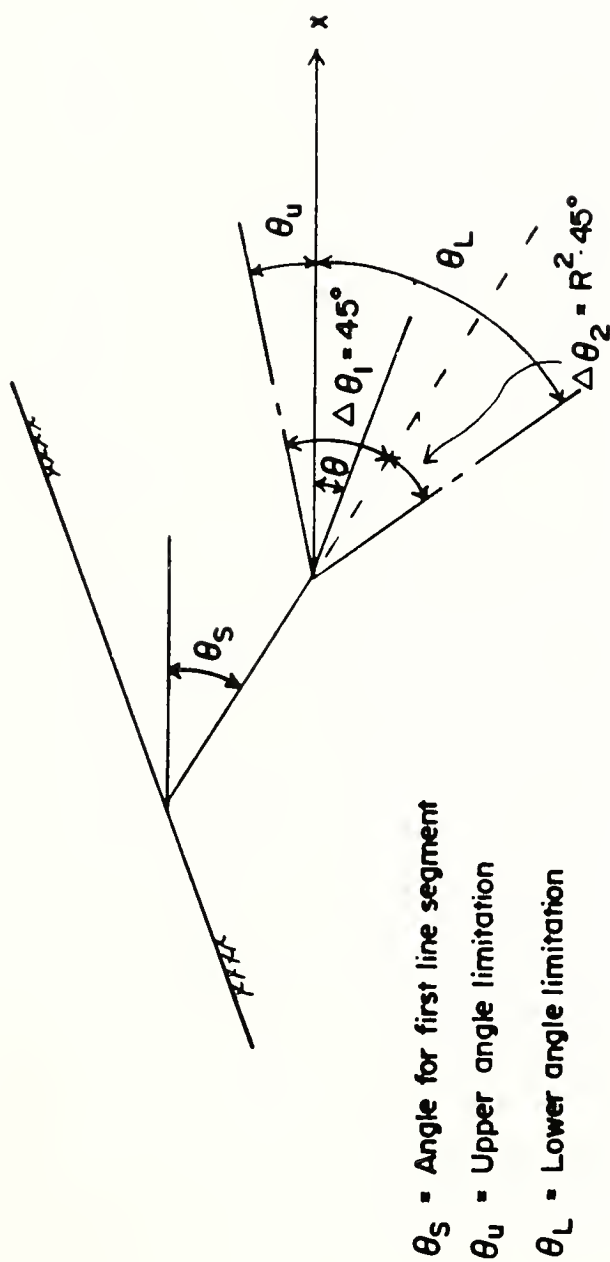


FIGURE D.2 ANGLE LIMITS FOR SUCCESSIVE LINE SEGMENTS OF IRREGULAR FAILURE SURFACE

MAIN PROGRAM STABL

READS AND DIRECTS EXECUTION OF COMMANDS.

INITIATES AND TERMINATES PLOTTING DEVICE.

COMMAND CODE	PROFIL	CALL	PROFIL	READS AND PRINTS PROBLEM TITLE.
			ENTRY	READS, CHECKS, STORES AND PRINTS SOIL PROFILE DATA.
SOIL	CALL		SOIL	READS, CHECKS, STORES AND PRINTS SOIL DATA.
			CALLS READER CALLS QUIT	- TO READ DATA IN FREE FORMAT. - IF DATA ARE FAULTY.
ANISO	CALL		ANISO	READS, CHECKS, STORES AND PRINTS ANISOTROPIC SOIL DATA.
			CALLS READER & QUIT	
WATER	CALL		WATER	READS, CHECKS, STORES AND PRINTS DATA FOR PIEZOMETRIC SURFACES.
			CALLS READER & QUIT	
LOADS	CALL		LOADS	READS, CHECKS, STORES AND PRINTS DATA FOR SURCHARGE.
			CALLS INTSC2 CALLS READER & QUIT	
EQUAKE	CALL		EQUAKE	READS, STORES AND PRINTS PSEUDO-STATIC EARTHQUAKE COEFFICIENTS AND CAVITATION PRESSURE.
			CALLS READER	
SURFAC SURBIS	CALL		SURFAC	READS, CHECKS, STORES AND PRINTS DATA FOR A SPECIFIED FAILURE SURFACE.
			CALLS READER & QUIT	
EXECUT (only after SURFAC or SURBIS)	CALL		SCALER	DETERMINES THE SCALE TO FIT PROBLEM GEOMETRY TO SIZE OF PLOT.
			PLTN	SETS UP AXIS FOR PRINT CHARACTER PLOT.
			EXECUT (see separate flow chart)	CONTROLS CALCULATION OF FACTOR OF SAFETY AND INTERMEDIATE STEPS. IF SINGLE TRIAL SURFACE IS SPECIFIED, IMPLEMENTS PLOT OF SURFACE AND PROBLEM GEOMETRY.
LIMITS	CALL		LIMITS	READS, CHECKS, STORES AND PRINTS DATA FOR LIMITATION BOUNDARIES FOR CRITICAL FAILURE SURFACE SEARCHING ROUTINES.
			CALLS READER & QUIT	
RANDOM CIRCLE2	CALL		RANDOM	READS, CHECKS, STORES AND PRINTS DATA FOR SEARCHING PARAMETERS.
			ENTRY	DIRECTS THE SEARCH AND ACCUMULATES DATA FOR THE TEN MOST CRITICAL FAILURE SURFACES, AND PRINTS THEM OUT.
BLOCK BLOCK2	CALL		BLOCK	
			(see separate flow chart)	
If faulty data	CALL		PLT4	BUILDS AND PRINTS PROBLEM GEOMETRY WITH PRINT CHARACTERS.
			CALLS POSTH	
	CALL		QUIT	STOPS EXECUTION OF STABL. TERMINATES PLOTTING DEVICE.

SUBROUTINE PLOTIN

PLOTS PROBLEM GEOMETRY WITH A PLOTTING DEVICE.

PLOTIN

ENTRY

PLOTN2

PLOTN3

PLOTS AXIS, PROFILE BOUNDARIES, PIEZOMETRIC SURFACES, IF ANY; LIMITATION BOUNDARIES, IF ANY; SURCHARGE LOADS, IF APPLICABLE.
PLOTS ALSO SEARCH BOXES FOR A SLIDING BLOCK SEARCH, IF APPLICABLE.

PLOTS INDIVIDUAL TRIAL FAILURE SURFACES.

PLOTS THE TEN MOST CRITICAL FAILURE SURFACES.

FIGURE D.3 FLOW CHART OF STABL

SUBROUTINE RANDOM, ENTRY BLOCK

READS, CHECKS, STORES AND PRINTS DATA FOR SEARCHING PARAMETERS.
DIRECTS THE SEARCH AND ACCUMULATES DATA FOR THE TEN MOST CRITICAL
FAILURE SURFACES, AND PRINTS THEM OUT.

COMMAND CODE CIRCLE CIRCLE2 RANDOM	CALL	READER (CALLS QUIT)	- TO READ DATA IN FREE FORMAT. - IF DATA ARE FAULTY.
	CALL	INTSC2 (see separate flow chart)	- TO CHECK IF A BOX IN THE SLIDING BLOCK SEARCH IS DEFINED TOTALLY BELOW THE GROUND SURFACE.
	CALL	SCALER	DETERMINES THE SCALE TO FIT PROBLEM GEOMETRY TO SIZE OF PLOT.
	CALL	RANSUF CALLS INTSC1 CALLS QUIT	GENERATES TRIAL FAILURE SURFACES OF CIRCULAR OR IRREGULAR SHAPE BY A RANDOM TECHNIQUE. - TO CALCULATE INTERSECTION POINT BETWEEN FAILURE SURFACE AND SOIL BOUNDARIES AND LIMITATION BOUNDARIES. - TO TERMINATE PROGRAM IF SURFACE POINT STORAGE LIMIT IS EXCEEDED.
BLOCK	CALL	BLKSUF CALLS INTSC1 CALLS INTSC2 CALLS QUIT	GENERATES TRIAL FAILURE SURFACES OF SLIDING BLOCK SHAPE BY A RANDOM TECHNIQUE. - see RANSUF above. - see RANSUF above.
BLOCK2	CALL	BLOCK2 CALLS INTSC1 CALLS INTSC2 CALLS QUIT	GENERATES TRIAL FAILURE SURFACES OF SLIDING BLOCK SHAPE BY A RANDOM TECHNIQUE. ACTIVE AND PASSIVE PORTIONS ARE GENERATED ACCORDING TO THE RANKINE THEORY. - TO CALCULATE INTERSECTION POINT BETWEEN FAILURE SURFACE AND SOIL BOUNDARIES. - LIKE BLKSUF, ALSO USED IN GENER- RATION OF ACTIVE AND PASSIVE PORTION OF FAILURE SURFACE. - see RANSUF above.
	CALL	EXECUT (see separate flow chart)	CONTROLS CALCULATION OF FACTOR OF SAFETY AND INTERMEDIATE STEPS.
	CALL	SORT	SORTS THE TEN MOST CRITICAL FAILURE SURFACES.
	CALL	PLOTIN (see separate flow chart)	PLOTS PROBLEM GEOMETRY WITH A PLOTING DEVICE.
	CALL	PLTN (see separate flow chart)	BUILDS AND PRINTS THE PRINT CHARACTER ARRAY CONTAINING THE PROBLEM GEOMETRY DISPLAYED WITH PRINT CHARACTERS.
	CALL	QUIT	- TO TERMINATE PROGRAM IF DATA ARE FAULTY OR IF ERRORS RC11 OR BK10 OCCUR.

SUBROUTINE PLTN

BUILDS AND PRINTS THE PRINT CHARACTER ARRAY CONTAINING THE PROBLEM
GEOMETRY DISPLAYED WITH PRINT CHARACTERS.

PLTN	SETS UP AXIS.
ENTRY	
PLT2	BUILDS DISPLAY OF INDIVIDUAL TRIAL FAILURE SURFACES.
PLT3	BUILDS DISPLAY OF THE TEN MOST CRITICAL FAILURE SURFACES.
PLT4	BUILDS DISPLAY OF THE PROBLEM GEOMETRY. PRINTS.
CALLS POSTH	DETERMINES THE POSITION OF A SCALED COORDINATE POINT WITHIN THE CHARACTER PLOT ARRAY.

FIGURE D.3 (cont.)

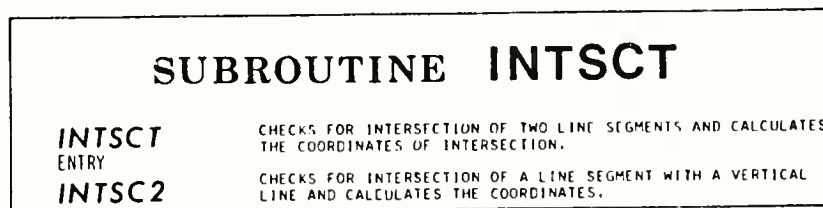
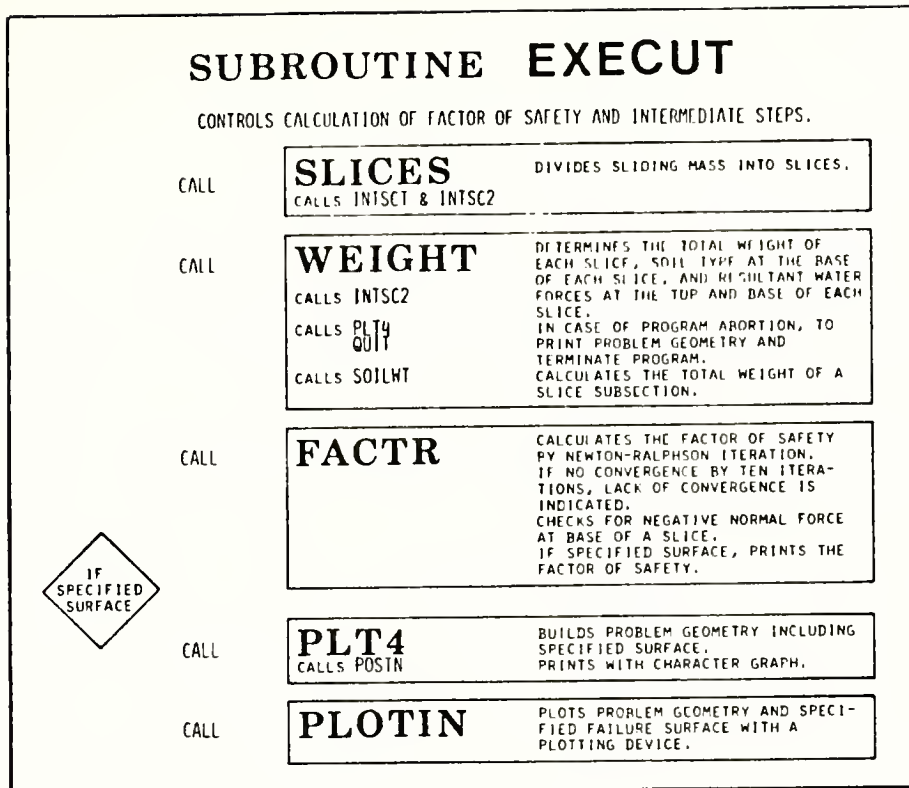


FIGURE D.3 (cont.)

APPENDIX E

LISTING OF REVISED STABL PROGRAM

A listing of the revised STABL program (STABL2) is enclosed on the following pages.

Marks in the margin of the listing indicates that the corresponding line (card) has been changed compared to the original STABL program. This was done in order that users, who have received copies of the STABL program at various times from the original STABL program was released in 1975, up to the present date (January, 1978), should be able to update their version of STABL, if they wish.

Some lines have been changed more than one time; however, only the final change appears in the listing of the revised program.

The symbols are as follows.

- (1) - Indicates change of that particular line.
- (2)  Indicates that all lines within the vertical line have been changed (including first and last line, where symbol (1) is used).
- (3)  Indicates that all lines between upper and lower arrow have been changed. The change may continue for several pages.

INDEX OF STABL2 PROGRAM

	<u>Page</u>
Main Program STABL2	394
Subroutine READER	399
Subroutine QUIT	403
Subroutine PROFIL	404
Subroutine ANISO.	414
Subroutine WATER.	418
Subroutine LOADS.	422
Subroutine EQUAKE	426
Subroutine LIMITS	427
Subroutine INTSCT	430
Subroutine SURFAC	432
Subroutine RANDOM	435
Subroutine RANSUF	451
Subroutine BLKSUF	463
Subroutine BLOCK2	468
Subroutine SORT	473
Subroutine EXECUT	474
Subroutine SLICES	475
Subroutine WEIGHT	480
Subroutine SOILWT	486
Subroutine FACTR.	488
Subroutine SCALER	493
Subroutine PLOTIN	495
Subroutine PLTN	501
Subroutine POSTN.	506

PROGRAM STABL2(INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT,PLOT)	STBL 2 -
-----	STBL 4
-----	STBL 6
PROGRAM STABL2	STBL 8 -
-----	STBL 10
-----	STBL 12
	STBL 14
AUTHOR -	STBL 16
	STBL 18
RONALD A. SIEGEL, GRADUATE RESEARCH ASSISTANT, 1975.	STBL 20 -
	STBL 22
REVISED BY -	OCT76EBO
	OCT76EBO
EVA BOUTRUP, GRADUATE RESEARCH ASSISTANT, JANUARY 1978.	JAN78EBO
	OCT76EBO
SPONSOR -	STBL 24
	STBL 26
JOINT HIGHWAY RESEARCH PROJECT	STBL 28
ENGINEERING EXPERIMENT STATION	STBL 30
PURDUE UNIVERSITY	STBL 32
IN COOPERATION WITH THE	STBL 34
INDIANA STATE HIGHWAY COMMISSION	STBL 36
	STBL 38
-----	STBL 40
-----	STBL 42
	STBL 44
FUNCTIONS -	STBL 46
	STBL 48
MAIN PROGRAM OF STABL.	STBL 50
	STBL 52
INITIATES PLOTTING DEVICE.	STBL 54
	STBL 56
READS AND DIRECTS EXECUTION OF COMMANDS.	STBL 58
	STBL 60
TERMINATES PLOTTING DEVICE.	STBL 62
	STBL 64
-----	STBL 66
	STBL 68
DEFINITIONS -	STBL 70
	STBL 72
ANISO SUBROUTINE THAT READS, STORES, AND PRINTS OUT	STBL 74
DIRECTIONAL STRENGTH PARAMETERS OF ANISOTROPIC SOIL	STBL 76
TYPES.	STBL 78
	STBL 80
BLOCK ENTRY OF SUBROUTINE RANDOM WHICH READS SEARCH	STBL 82
PARAMETERS AND DIRECTS RANDOM SEARCH FOR THE CRITICAL	STBL 84
SLIDING BLOCK SURFACE.	STBL 86
	STBL 88
EQUAKE SUBROUTINE THAT READS IN, STORES, AND PRINTS OUT	STBL 90
EARTHQUAKE DATA.	STBL 92
	STBL 94
ERROR VARIABLE REPRESENTING CURRENT ERROR CODE.	STBL 96
	STBL 98
EXECUT SUBROUTINE THAT DIRECTS CALCULATION OF THE FACTOR OF	STBL 100
SAFETY FOR A SPECIFIED FAILURE SURFACE.	STBL 102
	STBL 104
I INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	STBL 106
	STBL 108
IBLK CONTROL CODE WHICH ACTIVATES GENERATION OF A SLIDING	JUL76EBO
BLOCK SURFACE.	JUL76EBO
	JUL76EBO
IBLK2 CONTROL CODE WHICH ACTIVATES GENERATION OF A SLIDING	JUL76EBO
BLOCK SURFACE WITH RANKINE ACTIVE AND PASSIVE WEDGES.	JUL76EBO
	JUL76EBO
ICIRC CONTROL CODE WHICH DIRECTS GENERATION OF CIRCULAR OR	STBL 110
IRREGULAR SURFACES.	STBL 112
	STBL 114

C	SOIL	ENTRY OF SUBROUTINE PROFIL THAT READS IN, STORES,	STBL 216
C		AND PRINTS OUT ISOTROPIC SOIL PARAMETERS.	STBL 218
C			STBL 220
C	SURFAC	SUBROUTINE THAT READS IN, CHECKS, STORES, AND PRINTS	STBL 222
C		OUT DATA DEFINING AN INDIVIDUAL TRIAL FAILURE	STBL 224
C		SURFACE.	STBL 226
C			STBL 228
C	TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	STBL 230
C			STBL 232
C	WATER	SUBROUTINE THAT READS IN AND PRINTS OUT THE WATER	STBL 234
C		SURFACE DATA.	STBL 236
C			STBL 238
C		-----	STBL 240
C		-----	STBL 242
C			STBL 244
	COMMON /BLK01/ IANGL, IBLK, IEXIT, ICIRC, ILIMIT, IPLOT, IREAD, ISEARC,		STBL 246
	1 IBLK2, ISOIL, ISTR, ISURC, ISURF, IWAT, RD, TOL		STBL 248 -
	COMMON /BLK15/ M, MB		STBL 249 -
	DIMENSION KEYW(15), ERROR(5)		STBL 250 -
	DATA KEYW/6HPROFIL, 6HLOADS, 6HWATER, 6HSURFAC, 6HEXECUT, 6HEQUAKE,		STBL 252
	1 4HISOIL, 6HRANDOM, 6HCIRCLE, 6HCIRCL2, 6HBLOCK, 6HBLOCK2, 6HLIMITS,		STBL 254 -
	1 SHANISO, 6HSURBIS/		STBL 255 -
	DATA ERROR/4HSQ01, 4HSQ02, 4HSQ03, 4HSQ04, 4HSQ05/		STBL 256
	TOL=.0001		STBL 258 -
	RD=3.1415927/180.		STBL 260
			STBL 262
C	-----		STBL 264
C	INITIATE CONTROL CODES		STBL 266
C	-----		STBL 268
C			STBL 270
	IPLOT=0		STBL 272
	ICIRC=0		STBL 274
	ISEARC=0		STBL 276
	IEXIT=0		STBL 278
	IPIRF=0		STBL 280
	IBLK=0		STBL 281
	IBLK2=0		STBL 282
	MB=0		STBL 283
	N=0		STBL 284
			STBL 285
C	-----		STBL 286
C	INITIALIZE PLOTTING DEVICE		STBL 287
C	-----		STBL 288
	CALL PLOTS		STBL 290
	CALL PLOT(0., 1.4, -2)		STBL 292
	CALL PLOT(2.5, 0., -3)		STBL 294
			STBL 296
C	-----		STBL 298
C	READ COMMAND CARD		STBL 300
C	-----		STBL 302
	11 IREAD=0		STBL 304
	READ(5, 100) MKEYW		STBL 306
	100 FORMAT(A6)		STBL 308
			STBL 310
			STBL 312
			STBL 314
C	-----		STBL 316
C	CHECK IF READING BEYOND END OF FILE		STBL 318
C	-----		STBL 320
	IF(EOF, 5) 19, 50		STBL 322
			STBL 324
C			STBL 326

C	-----	STBL 328
C	CHECK IF PROFIL IS INITIAL COMMAND	STBL 330
C	-----	STBL 332
	50 IF(IPROF.EQ.1)GO TO 23	STBL 334
	IF(MKEYH.EQ.KEYH(1))GO TO 24	STBL 336
	WRITE(6,106)ERROR(1)	STBL 338
	106 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****, /)	STBL 340
	IEXIT=1	STBL 342
	24 IPROF=1	STBL 344
		STBL 346
C	-----	STBL 348
C	MATCH COMMAND WITH A DIRECTIVE AND EXECUTE IT	STBL 350
C	-----	STBL 352
C		STBL 354
	23 DO 12 I=1,15	STBL 356
	IF(MKEYH.EQ.KEYH(I))GO TO (1,2,3,4,5,6,7,9,10,8,14,18,15,16,13),I	STBL 358 -
	12 CONTINUE	STBL 360 -
		STBL 362
C	-----	STBL 364
C	TERMINATE EXECUTION IF ILLEGAL COMMAND ENCOUNTERED	STBL 366
C	-----	STBL 368
	WRITE(6,101)MKEYH	STBL 370
	101 FORMAT(1H1, /	STBL 372
	1 10X, ***** /	STBL 374
	1 10X, ***** ,A6, - ILLEGAL COMMAND ***** /	STBL 376
	1 10X, ***** /	STBL 378
	WRITE(6,102)(KEYH(I),I=1,15)	STBL 380
	102 FORMAT(/,10X,LEGAL COMMANDS - ,A6, /, (27X,A6))	STBL 382
	CALL QUIT	STBL 384 -
	1 CALL PROFIL	STBL 386
	GO TO 11	STBL 388
	2 CALL LOADS	STBL 390
	GO TO 11	STBL 392
	3 CALL WATER	STBL 394
	GO TO 11	STBL 396
	13 MB=1	STBL 398
	N=0	STBL 400
	4 CALL SURFAC	STBL 401 -
	IF(N.EQ.1) MB=0	STBL 402 -
	GO TO 11	STBL 403
		STBL 404 -
		STBL 405
C	-----	STBL 406
C	CHECK EXECUT SEQUENCING	STBL 408
C	-----	STBL 410
C		STBL 412
	5 IF(ISOIL.EQ.1)GO TO 20	STBL 414
	WRITE(6,106)ERROR(2)	STBL 416
	CALL QUIT	STBL 418
	20 IF(ISURF.EQ.1)GO TO 21	STBL 420
	WRITE(6,106)ERROR(3)	STBL 422
	CALL QUIT	STBL 424
	21 CALL SCALER	STBL 426
	CALL PLTN	STBL 428
	IF(IEXIT.EQ.0)GO TO 17	STBL 430
		STBL 432
C	-----	STBL 434
C	TERMINATE EXECUTION IF DATA FAULTY	STBL 436
C	-----	STBL 438
C		STBL 440
	CALL PLT4	STBL 442
	CALL QUIT	STBL 444
	17 CALL EXECUT	STBL 446
	N=1	STBL 448
	GO TO 11	STBL 449 -
	6 CALL EQUAKE	STBL 450
	GO TO 11	STBL 452
		STBL 454

7 CALL SOIL	STBL 456
GO TO 11	STBL 458
9 IF(ISOIL.EQ.1)GO TO 25	STBL 460
WRITE(6,106)ERROR(5)	STBL 462
25 CALL RANDOM	STBL 464
GO TO 11	STBL 466
8 MB=1	STBL 467 -
N=0	STBL 468 -
10 ICIRC=1	STBL 469
IF(N.EQ.1) MB=0	STBL 470 -
IF(ISOIL.EQ.1)GO TO 26	STBL 471
WRITE(6,106)ERROR(5)	STBL 472
26 CALL RANDOM	STBL 474
GO TO 11	STBL 476
14 IF(ISOIL.EQ.1)GO TO 27	STBL 477
WRITE(6,106)ERROR(5)	STBL 478
18 IBLK2=1	STBL 479 -
IF(ISOIL.EQ.1)GO TO 27	STBL 480 -
WRITE(6,106)ERROR(5)	STBL 481 -
27 CALL BLOCK	STBL 482
GO TO 11	STBL 484
15 CALL LIMITS	STBL 486
GO TO 11	STBL 488
C	STBL 490
C	STBL 492
C	STBL 494
C	STBL 496
C	STBL 498
16 IF(ISOIL.EQ.1)GO TO 22	STBL 500
WRITE(6,106)ERROR(4)	STBL 502
IEXIT=1	STBL 504
22 CALL ANISO	STBL 506
GO TO 11	STBL 508
C	STBL 510
C	STBL 512
C	STBL 514
C	STBL 516
C	STBL 518
19 CALL PLOT(0.,0.,999)	STBL 520
STOP	STBL 522
END	STBL 524


```

C      SUM      REAL NUMBER THAT IS ASSEMBLED.      READ 136
C      -----      READ 138
C      -----      READ 140
C      -----      READ 142
C      -----      READ 144
COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPL0T,IREAD,ISEARC,
1      IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL      READ 146
COMMON /BLK02/NSLICE,X(300)      READ 148 -
DIMENSION M(80),N(14),A(10),IA(10),ERROR(7)      READ 150
EQUIVALENCE (M(1),X(1))      READ 152
DATA N/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H ,1H.,1H0,1H-/      READ 154
DATA A/1.,2.,3.,4.,5.,6.,7.,8.,9.,0./      READ 156
DATA IA/1,2,3,4,5,6,7,8,9,0/      READ 158
DATA ERROR/4HFR01,4HFR02,4HFR03,4HFR04,4HFR05,4HFR06,4HFR07/      READ 160
SSIGN=1.      READ 162
IF(IREAD.EQ.1.AND.IR.EQ.1)GO TO 3      READ 164
IF(IREAD.EQ.1)GO TO 11      READ 166
C      -----      READ 170
C      -----      READ 172
C      -----      READ 174
C      -----      READ 176
C      -----      READ 178
5 READ(5,101)(M(I),I=1,80)      READ 180
101 FORMAT(80A1)      READ 182
C      -----      READ 184
C      -----      READ 186
C      -----      READ 188
C      -----      READ 190
C      -----      READ 192
IF(E0F,5)23,24      READ 194
23 WRITE(6,105)ERROR(1)      READ 196
105 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,/)      READ 198
CALL QUIT      READ 200
24 IF(M(1).NE.N(11))GO TO 22      READ 202
WRITE(6,105)ERROR(2)      READ 204
WRITE(6,107)(M(II),II=1,80)      READ 206
107 FORMAT(1X,*,**** ERROR IN FOLLOWING LINE OF DATA *****,//,
1      1X,1H$,/,1X,80A1,/)      READ 208
C      -----      READ 210
C      -----      READ 212
C      -----      READ 214
C      -----      READ 216
C      -----      READ 218
C      -----      READ 220
CALL QUIT      READ 222
22 IREAD=1      READ 224
K=1      READ 226
I=0      READ 228
IF(IR.EQ.1)GO TO 4      READ 230
GO TO 12      READ 232
C      -----      READ 234
C      -----      READ 236
C      -----      READ 238
C      -----      READ 240
C      -----      READ 242
11 IF(M(I).EQ.N(11).AND.M(I+1).EQ.N(11))GO TO 5      READ 244
12 I=I+1      READ 246
C      -----      READ 248
C      -----      READ 250
C      -----      READ 252
C      -----      READ 254
C      -----      READ 256
IF(M(I).EQ.N(13))M(I)=N(10)      READ 258
C      -----      READ 260

```


C	-----	READ 262
C	CHECK FOR DECIMAL POINT	READ 264
C	-----	READ 266
C	IF(M(I).NE.N(12))GO TO 16	READ 268
	WRITE(6,105)ERROR(3)	READ 270
	IEXIT=2	READ 272
C	-----	READ 274
C	CHECK SIGN OF INTEGER	READ 276
C	-----	READ 278
C	16 IF(M(I).NE.N(14))GO TO 20	READ 280
	WRITE(6,105)ERROR(4)	READ 282
	IEXIT=2	READ 284
C	-----	READ 286
C	TERMINATE EXECUTION IF DATA FAULTY	READ 288
C	-----	READ 290
C	20 IF(M(I).NE.N(11))GO TO 12	READ 292
	IF(IEXIT.LE.1)GO TO 26	READ 294
	WRITE(6,107)(M(II),II=1,80)	READ 296
	CALL QUIT	READ 298
	26 KI=K	READ 300
	K=I+1	READ 302
	KF=I-1	READ 304
	ISUM=0	READ 306
C	-----	READ 308
C	DETERMINE DIGITS OF INTEGER NUMBER	READ 310
C	-----	READ 312
C	DO 13 II=KI,KF	READ 314
	DO 14 J=1,10	READ 316
	L=J	READ 318
	IF(M(II).EQ.N(J))GO TO 15	READ 320
	14 CONTINUE	READ 322
C	-----	READ 324
C	TERMINATE EXECUTION IF DATA FAULTY	READ 326
C	-----	READ 328
C	WRITE(6,105)ERROR(5)	READ 330
	WRITE(6,107)(M(IJ),IJ=1,80)	READ 332
	CALL QUIT	READ 334
	15 ISUM=ISUM+IA(L)*10**(KF-II)	READ 336
	13 CONTINUE	READ 338
	RETURN	READ 340
C	-----	READ 342
C	ASSEMBLE REAL NUMBER	READ 344
C	-----	READ 346
C	3 IF(M(I).EQ.N(11).AND.M(I+1).EQ.N(11))GO TO 5	READ 348
	4 I=I+1	READ 350
C	-----	READ 352
C	IF LETTER (O) READ, SUBSTITUTE NUMBER (0)	READ 354
C	-----	READ 356
C	IF(M(I).EQ.N(13))M(I)=N(10)	READ 358
	IF(M(I).EQ.N(12))KD=I	READ 360
	IF(M(I).NE.N(11))GO TO 4	READ 362
	KI=K	READ 364
	K=I+1	READ 366
C		READ 368
		READ 370
		READ 372
		READ 374
		READ 376
		READ 378
		READ 380
		READ 382
		READ 384
		READ 386
		READ 388
		READ 390
		READ 392
		READ 394


```

C -----
C CHECK SIGN
C -----
C
C     IF(M(KI).NE.N(14))GO TO 19
C     KI=KI+1
C     SSIGN=-1.
19  KF=I
C     KD1=KD-1
C     SUM=0.
C
C -----
C CHECK FOR DECIMAL POINT
C -----
C
C     DO 17 II=KI,KF
C     IF(M(II).EQ.N(12))GO TO 18
17  CONTINUE
C     WRITE(6,105)ERROR(6)
C
C -----
C CHECK FOR ILLEGAL CHARACTERS
C -----
C
C     DO 25 II=KI,KF
C     DO 21 J=1,14
C     IF(M(II).EQ.N(J))GO TO 25
21  CONTINUE
C     WRITE(6,105)ERROR(7)
25  CONTINUE
C     WRITE(6,107)(M(II),II=1,80)
C
C -----
C TERMINATE EXECUTION IF DATA FAULTY
C -----
C
C     CALL QUIT
C
C -----
C DETERMINE DIGITS OF REAL NUMBER
C -----
C
18  DO 2 II=KI,KF
C     IF(II.GE.KD)GO TO 6
C     DO 7 J=1,10
C     L=J
C     IF(M(II).EQ.N(J))GO TO 8
7   CONTINUE
C
C -----
C TERMINATE EXECUTION IF DATA FAULTY
C -----
C
C     WRITE(6,105)ERROR(7)
C     WRITE(6,107)(M(IJ),IJ=1,80)
C     CALL QUIT
8   SUM=SUM+A(L)*10.** (KD1-II)
C     GO TO 2
6   IF(II.EQ.KD)GO TO 2
C     DO 9 J=1,10
C     L=J
C     IF(M(II).EQ.N(J))GO TO 10
9   CONTINUE
10  SUM=SUM+A(L)/10.** (II-KD)
2   CONTINUE
C

```

```

READ 396
READ 398
READ 400
READ 402
READ 404
READ 406
READ 408
READ 410
READ 412
READ 414
READ 416
READ 418
READ 420
READ 422
READ 424
READ 426
READ 428
READ 430
READ 432
READ 434
READ 436
READ 438
READ 440
READ 442
READ 444
READ 446
READ 448
READ 450
READ 452
READ 454
READ 456
READ 458
READ 460
READ 462
READ 464
READ 466
READ 468
READ 470
READ 472
READ 474
READ 476
READ 478
READ 480
READ 482
READ 484
READ 486
READ 488
READ 490
READ 492
READ 494
READ 496
READ 498
READ 500
READ 502
READ 504
READ 506
READ 508
READ 510
READ 512
READ 514
READ 516
READ 518
READ 520
READ 522
READ 524
READ 526

```


C	-----	READ	528
C	ACCOUNT FOR SIGN	READ	530
C	-----	READ	532
C		READ	534
	SUM=SUM*SSIGN	READ	536
	RETURN	READ	538
	END	READ	540

	SUBROUTINE QUIT	QUIT	2	
C	-----	QUIT	4	
C		QUIT	6	
C	SUBROUTINE QUIT	QUIT	8	
C	-----	QUIT	10	
C		QUIT	12	
C		QUIT	14	
C	FUNCTIONS -	QUIT	16	
C		QUIT	18	
C	STOPS EXECUTION OF STABL.	QUIT	20	
C		QUIT	22	
C	TERMINATES PLOTTING DEVICE.	QUIT	24	
C		QUIT	26	
C	-----	QUIT	28	
C		QUIT	30	
C	DEFINITION -	QUIT	32	
C		QUIT	34	
C	PLOT	EXTERNAL SUBROUTINE USED TO MOVE A PLOTTING PEN TO A	QUIT	36
C		NEW POINT. 999 FOR THIRD ARGUMENT TERMINATES	QUIT	38
C		PLOTTING.	QUIT	40
C			QUIT	42
C	-----	QUIT	44	
C		QUIT	46	
		QUIT	48	
	WRITE(6,102)	QUIT	50	
102	FORMAT(//,	QUIT	52	
	110X*****//	QUIT	54	
	110X***** EXECUTION OF STABL ABORTED *****//	QUIT	56	
	110X*****//)	QUIT	58	
	CALL PLOT(0.,0.,999)	QUIT	60	
	STOP	QUIT	62	
	END	QUIT	64	

C C C C C C C C C C C C C C C	SUBROUTINE PROFIL	PROF 2
	-----	PROF 4
	-----	PROF 6
	SUBROUTINE PROFIL	PROF 8
	-----	PROF 10
	-----	PROF 12
		PROF 14
	FUNCTIONS -	PROF 16
		PROF 18
	INITIALIZES CONTROL CODES FOR PROBLEM.	PROF 20
		PROF 22
	PRINTS TITLE.	PROF 24
		PROF 26
	READS AND PRINTS PROBLEM DESCRIPTION.	PROF 28
		PROF 30
C C C C C C C C C C C C C C C	READS PROFILE GEOMETRY DATA DEFINING THE GROUND SURFACE AND THE	PROF 32
	SUBSURFACE INTERFACES BETWEEN DIFFERING SOIL TYPES.	PROF 34
		PROF 36
	CHECKS DATA FOR SATISFACTION OF PROGRAM REQUIREMENTS.	PROF 38
		PROF 40
	STORES DATA IN BLOCK COMMON.	PROF 42
		PROF 44
	PRINTS PROFILE GEOMETRY DATA.	PROF 46
		PROF 48
	ENTRY SOIL *****	PROF 50
		PROF 52
	READS THE NUMBER OF SOIL TYPES, ZERO INDICATES MODIFICATION TO	PROF 54
	BE MADE TO EXISTING SOIL DATA.	PROF 56
		PROF 58
	CHECKS FOR SOIL DATA MODIFICATION OR TOTAL DEFINITION.	PROF 60
C C C C C C C C C C C C C C C		PROF 62
	IF MODIFICATION, READS, CHECKS, AND STORES CHANGES, THEN PRINTS	PROF 64
	ALL SOIL DATA.	PROF 66
		PROF 68
	IF TOTAL DEFINITION, READS, CHECKS, STORES, AND PRINTS ALL SOIL	PROF 70
	DATA.	PROF 72
		PROF 74
	-----	PROF 76
	DEFINITIONS -	PROF 78
		PROF 80
		PROF 82
	ABS STANDARD FUNCTION WHICH DETERMINES THE ABSOLUTE	PROF 84
	VALUE OF A NUMBER.	PROF 86
		PROF 88
	AI INTERCEPT OF BOUNDARY (I).	PROF 90
		PROF 92
	AJ INTERCEPT OF BOUNDARY (J).	PROF 94
		PROF 96
	BI SLOPE OF BOUNDARY (I).	PROF 98
		PROF 100
	BJ SLOPE OF BOUNDARY (J).	PROF 102
		PROF 104
	BNDS AN ARRAY CONTAINING THE COORDINATES OF THE END POINTS	PROF 106
	DEFINING THE GROUND SURFACE AND SUBSURFACE	PROF 108
	BOUNDARIES.	PROF 110
		PROF 112
	C ARRAY CONTAINING ISOTROPIC VALUES OF THE MOHR-COULOMB	PROF 114
	COHESION FOR EACH SOIL TYPE.	PROF 116
		PROF 118
	CU ARRAY CONTAINING VALUES OF PORE PRESSURE CONSTANTS FOR	PROF 120
	EACH SOIL TYPE.	PROF 122
		PROF 124
	DENOI DENOMINATOR FOR SLOPE COMPUTATION OF BOUNDARY (I).	PROF 126
		PROF 128
	DENOJ DENOMINATOR FOR SLOPE COMPUTATION OF BOUNDARY (J).	PROF 130
		PROF 132
	DESCR ARRAY CONTAINING THE SECOND LINE OF THE PROBLEM	PROF 134 -
	DESCRIPTION.	PROF 135 -

C			PROF 136
C	DUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN	PROF 138
C		AN INTEGER IS TO BE READ.	PROF 140
C			PROF 142
C	ERROR	VARIABLE REPRESENTING CURRENT ERROR CODE.	PROF 144
C			PROF 146
C	GAMMA	ARRAY CONTAINING VALUES OF THE TOTAL UNIT WEIGHT FOR	PROF 148
C		EACH SOIL TYPE.	PROF 150
C			PROF 152
C	GSAT	ARRAY CONTAINING VALUES OF THE SATURATED UNIT WEIGHT	PROF 154
C		FOR EACH SOIL TYPE.	PROF 156
C			PROF 158
C	I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	PROF 160
C			PROF 162
C	IDUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN	PROF 164
C		A REAL NUMBER IS TO BE READ.	PROF 166
C			PROF 168
C	IEXIT	CONTROL CODE WHICH TERMINATES EXECUTION, IF DATA IS	PROF 170
C		INCONSISTANT WITH THE REQUIREMENTS OF THE PROGRAM.	PROF 172
C			PROF 174
C	II	VARIABLE USED AS DO LOOP INITIATION VALUE.	PROF 176
C			PROF 178
C	ILIMIT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	PROF 180
C		AFFECTED BY SEARCHING LIMITS ESTABLISHED BY	PROF 182
C		SUBROUTINE LIMITS.	PROF 184
C			PROF 186
C	IPLOT	CONTROL CODE WHICH CONTROLS TRANSLATION OF AXES FOR	NOU76EBO -
C		PLOTTED OUTPUT.	NOU76EBO -
C			NOU76EBO -
C	ISOIL	CONTROL CODE WHICH INDICATES DEFINITION OF ISOTROPIC	PROF 188
C		SOIL DATA.	PROF 190
C			PROF 192
C	ISTR	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	PROF 194
C		HANDLING ANISOTROPIC STRENGTH CONDITION.	PROF 196
C			PROF 198
C	ISURC	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	PROF 200
C		HANDLING THE SURCHARGE LOADS SPECIFIED BY SUBROUTINE	PROF 202
C		LOADS.	PROF 204
C			PROF 206
C	ISURF	CONTROL CODE WHICH INDICATES DEFINITION OF SPECIFIED	PROF 208
C		TRIAL FAILURE SURFACE.	PROF 210
C			PROF 212
C	ITP	ARRAY CONTAINING SOIL TYPE INDICES FOR EACH BOUNDARY.	PROF 214
C			PROF 216
C	IWAT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	PROF 218
C		HANDLING THE WATER SURFACE SPECIFIED BY SUBROUTINE	PROF 220
C		WATER.	PROF 222
C			PROF 224
C	J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	PROF 226
C			PROF 228
C	K	SOIL TYPE WHICH IS TO HAVE ISOTROPIC PARAMETERS	PROF 230
C		CHANGED.	PROF 232
C			PROF 234
C	KCOEF	HORIZONTAL EARTHQUAKE COEFFICIENT.	PROF 236
C			PROF 238
C	NBND	TOTAL NUMBER OF PROFILE BOUNDARIES.	PROF 240
C			PROF 242
C	NBND1	NBND - 1	PROF 244
C			PROF 246
C	NCHA	NUMBER OF SOIL TYPE CHANGES.	PROF 248
C			PROF 250
C	NLIMIT	NUMBER OF SURFACE GENERATION LIMIT BOUNDARIES.	PROF 252
C			PROF 254
C	NP	ARRAY CONTAINING NUMBER OF PIEZOMETRIC SURFACE	JUL76EBO -
C		FOR EACH SOIL TYPE.	JUL76EBO -
C			JUL76EBO -
C	NPZ	NUMBER OF PIEZOMETRIC SURFACES DEFINED.	PROF 256
C			PROF 258

NSAL	NUMBER OF SOIL TYPES HAVING ANISOTROPIC STRENGTH PROPERTIES.	PROF 260
		PROF 262
		PROF 264
NSOI	NUMBER READ AS NUMBER OF SOIL TYPES.	PROF 266
		PROF 268
NSOIL	NUMBER OF SOIL TYPES SPECIFIED.	PROF 270
		PROF 272
NSURC	NUMBER OF BOUNDARY LOADS SPECIFIED.	PROF 274
		PROF 276
NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	PROF 278
		PROF 280
PHI	ARRAY CONTAINING ISOTROPIC VALUES OF THE MOHR-COULOMB ANGLE OF INTERNAL FRICTION FOR EACH SOIL TYPE.	PROF 286 *
		PROF 288
		PROF 290
PLOT	EXTERNAL SUBROUTINE USED TO MOVE A PLOTTING PEN TO A NEW POINT.	NOU76EBO -
		NOU76EBO -
		NOU76EBO -
QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE, TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE PROGRAM.	PROF 292
		PROF 294
		PROF 296
		PROF 298
READER	SUBROUTINE THAT READS INTEGER OR REAL DATA IN FREE FORM FORMAT.	PROF 300
		PROF 302
		PROF 304
RU	ARRAY CONTAINING VALUES OF THE PORE PRESSURE PARAMETER FOR EACH SOIL TYPE.	PROF 306
		PROF 308
		NOU75EBO
SYMBOL	EXTERNAL SUBROUTINE THAT ACTIVATES THE PLOTTING PEN TO PLOT CHARACTERS OR ON-CENTER SYMBOLS.	NOU76EBO
		NOU76EBO
		NOU76EBO
TITLE	ARRAY CONTAINING THE FIRST LINE OF THE PROBLEM DESCRIPTION.	NOU75EBO
		NOU75EBO
		NOU75EBO
TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	PROF 310
		PROF 312
UKCOEF	VERTICAL EARTHQUAKE COEFFICIENT.	PROF 314
		PROF 316
XM	X1 - TOL	PROF 318
		PROF 320
XP	X1 + TOL	PROF 322
		PROF 324
X1	X COORDINATE OF THE PROJECTED INTERSECTION OF TWO BOUNDARY LINE SEGMENTS.	PROF 326
		PROF 328
		PROF 330
Y1	Y COORDINATE OF THE PROJECTED INTERSECTION OF TWO BOUNDARY LINE SEGMENTS.	PROF 332
		PROF 334
		PROF 336
		PROF 338
		PROF 340
		PROF 342
	DIMENSION TITLE(4), DESCR(4), ERROR(11)	PROF 344 -
	COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLOT,IREAD,ISEARC,	PROF 346
1	IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL	PROF 348 -
	COMMON /BLK02/BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND,	PROF 350
1	NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20)	PROF 352 -
	COMMON /BLK03/UWAT,NP2,NPIEZ(10),XPIEZ(10,40),YPIEZ(10,40)	PROF 354 -
	COMMON /BLK04/DELTA(10),LOAD(10),NSURC,SURC(10,2),YSURC(10,2)	PROF 356
	COMMON /BLK06/LIMIT(20,4),NLIMIT,NLMT	PROF 358
	COMMON /BLK07/CSA(10,10),DIREC(10,10),ITPA(10),NDIREC(10),NSAL,	PROF 360 -
1	PHIA(10,10)	PROF 362
	COMMON /BLK11/CAUT,KCOEF,UKCOEF	PROF 364
	DATA ERROR/4HPF01,4HPF02,4HPF03,4HPF04,4HPF05,4HPF06,4HSL01,	PROF 366
1	4HSL02,4HSL03,4HSL04,4HSL05/	PROF 368 *
	REAL KCOEF	PROF 372
	KCOEF=0.	PROF 374
	UKCOEF=0.	PROF 376
	NSOIL=0	PROF 378

* Mark between lines indicates that lines have been removed.

	NPZ=0	PROF 380 -
	NSURC=0	PROF 382
	NLIMIT =0	PROF 384
	NSAL=0	PROF 386
C		PROF 388
C	-----	PROF 390
C	INITIALIZE CONTROL CODES	PROF 392
C	-----	PROF 394
	ISTR=0	PROF 396
	IWAT=0	PROF 398
	ISURC=0	PROF 400
	ILIMIT=0	PROF 402
	ISOIL=0	PROF 404
	ISURF=0	PROF 406
	IF(IPLT.EQ.0) GO TO 28	PROF 408
C		PROF 409
C	-----	PROF 410
C	IF NOT, MAKE PAGE MARK AND POSITION NEW ORIGIN	PROF 411
C	-----	PROF 412
	CALL PLOT(14.0,0.,-3)	PROF 413
	CALL PLOT(-2.5,0.,3)	PROF 414
	CALL PLOT(-2.5,-1.4,2)	PROF 415
	IPLT=0	PROF 416
		PROF 417
C		PROF 418
C	-----	PROF 419
C	PRINT TITLE	PROF 420
C	-----	PROF 421
	28 WRITE(6,111)	PROF 422
	111 FORMAT(1H1,23X,---SLOPE STABILITY ANALYSIS---,/	PROF 423
	1 21X,---SIMPLIFIED JANBU METHOD OF SLICES,/	PROF 424 -
	1 25X,---IRREGULAR FAILURE SURFACES---/)	PROF 425
C		PROF 426 -
C	-----	PROF 427
C	READ AND PRINT PROBLEM DESCRIPTION	PROF 428
C	-----	PROF 430
	READ(5,113) TITLE, DESCR	PROF 432
	113 FORMAT(8A10)	PROF 434
	WRITE(6,112) TITLE, DESCR	PROF 436
	112 FORMAT(//,10X,---PROBLEM DESCRIPTION ---,4A10,/32X,4A10,//)	PROF 438
	CALL SYMBOL (0.,7.5,0.14,TITLE,0.,40)	PROF 440
	CALL SYMBOL (0.,7.2,0.14,DESCR,0.,40)	PROF 442
		PROF 443
C		PROF 444
C	-----	PROF 445
C	READ NUMBER OF BOUNDARIES DATA	PROF 446
C	-----	PROF 448
	CALL READER(DUMMY,NBND,0)	PROF 450
	CALL READER(DUMMY,NTOP,0)	PROF 452
		PROF 454
C		PROF 456
C	-----	PROF 458
C	PRINT NUMBER OF BOUNDARIES DATA	PROF 460
C	-----	PROF 462
	WRITE(6,107)NTOP,NBND	PROF 464
	107 FORMAT(//,10X,---BOUNDARY COORDINATES---,//,	PROF 466
	1 12X,I3,--- TOP BOUNDARIES---,/,	PROF 468
	1 12X,I3,--- TOTAL BOUNDARIES---)	PROF 470
C		PROF 472
		PROF 474
		PROF 476
		PROF 478


```

C -----
C CHECK IF NUMBER OF GROUND SURFACE BOUNDARIES IS LESS
C THAN OR EQUAL TO THE TOTAL NUMBER OF BOUNDARIES
C -----
C     IF(NBND.GE.NTOP)GO TO 16
C     WRITE(6,105)ERROR(1)
C 105 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,/)
C     IEXIT=1
C
C -----
C CHECK BOUNDARY STORAGE LIMIT
C -----
C 16 IF(NBND.LE.100)GO TO 20
C     WRITE(6,105)ERROR(2)
C     CALL QUIT
C
C -----
C PRINT BOUNDARY DATA HEADINGS
C -----
C 20 WRITE(6,119)
C 119 FORMAT(/,
C 1 10X,=BOUNDARY=,5X,=X-LEFT=,5X,=Y-LEFT=,4X,=X-RIGHT=,4X,
C 1 =Y-RIGHT=,4X,=SOIL TYPE=,/,
C 1 13X,=NO.=,8X,=(FT)=,7X,=(FT)=,7X,=(FT)=,7X,=(FT)=,5X,
C 1 =BELOW BND=,/)
C
C -----
C READ BOUNDARY DATA, CHECK FOR 1ST QUADRANT
C PROFILE, AND CHECK BOUNDARY END POINT ORDER
C -----
C     DO 4 I=1,NBND
C     DO 2 J=1,4
C     CALL READER(BNDS(I,J),IDUMMY,1)
C     IF(BNDS(I,J).GT.-TOL)GO TO 2
C     WRITE(6,104)ERROR(3),I
C 104 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,5X,8HBOUNDARY,I3,/)
C     IEXIT=1
C     2 CONTINUE
C     IF((BNDS(I,1)-BNDS(I,3)).LT.TOL)GO TO 3
C     WRITE(6,104)ERROR(4),I
C     IEXIT=1
C     3 CALL READER(DUMMY,ITP(I),0)
C     4 CONTINUE
C     IF(NTOP.EQ.1)GO TO 14
C
C -----
C CHECK GROUND SURFACE BOUNDARY ORDER
C -----
C     DO 10 I=2,NTOP
C     IF(ABS(BNDS(I-1,3)-BNDS(I,1)).LT.TOL.AND.ABS(BNDS(I-1,4)-BNDS(I,2)
C 1).LT.TOL)GO TO 10
C     J=I-1
C     WRITE(6,103)ERROR(5),J,I
C 103 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,5X,10HBOUNDARIES,I3,
C 1 4H AND,I3,/)
C     IEXIT=1
C 10 CONTINUE
C

```

PROF 480
 PROF 482
 PROF 484
 PROF 486
 PROF 488
 PROF 490
 PROF 492
 PROF 494
 PROF 496
 PROF 498
 PROF 500
 PROF 502
 PROF 504
 PROF 506
 PROF 508
 PROF 510
 PROF 512
 PROF 514
 PROF 516
 PROF 518
 PROF 520
 PROF 522
 PROF 524
 PROF 526
 PROF 528
 PROF 530
 PROF 532
 PROF 534
 PROF 536
 PROF 538
 PROF 540
 PROF 542
 PROF 544
 PROF 546
 PROF 548
 PROF 550
 PROF 552
 PROF 554
 PROF 556
 PROF 558
 PROF 560
 PROF 562
 PROF 564
 PROF 566
 PROF 568
 PROF 570
 PROF 572
 PROF 574
 PROF 576
 PROF 578
 PROF 580
 PROF 582
 PROF 584
 PROF 586
 PROF 588
 PROF 590
 PROF 592
 PROF 594
 PROF 596
 PROF 598
 PROF 600
 PROF 602
 PROF 604

C	-----	PROF 606
C	CHECK SUBSURFACE BOUNDARY ORDER	PROF 608
C	-----	PROF 610
C	14 IF (NTOP.EQ.NBND)GO TO 5	PROF 612
	NBND1=NBND-1	PROF 614
	DO 8 I=NTOP,NBND1	PROF 618
	II=I+1	PROF 620
	DO 8 J=II,NBND	PROF 622
C	-----	PROF 624
C	CHECK FOR OVERLAP OF HORIZONTAL PROJECTIONS	PROF 626
C	OF BOUNDARIES (I) AND (J)	PROF 628
C	-----	PROF 630
C	IF((BNDS(J,1).GE.BNDS(I,3)).OR.(BNDS(J,3).LE.BNDS(I,1)))GO TO 8	PROF 632
	DENOI=BNDS(I,1)-BNDS(I,3)	PROF 634
	DENOJ=BNDS(J,1)-BNDS(J,3)	PROF 636
C	-----	PROF 638
C	CHECK IF BOUNDARY (I) IS VERTICAL	PROF 640
C	-----	PROF 642
C	IF (ABS(DENOI).GE.TOL)GO TO 7	PROF 644
C	-----	PROF 646
C	CHECK IF BOUNDARY (J) IS VERTICAL	PROF 648
C	-----	PROF 650
C	IF (ABS(DENOJ).GE.TOL)GO TO 9	PROF 652
C	-----	PROF 654
C	CHECK IF VERTICAL BOUNDARY (I) IS ABOVE VERTICAL BOUNDARY (J)	PROF 656
C	-----	PROF 658
C	IF((BNDS(J,2)-BNDS(I,2)).LT.TOL.AND.(BNDS(J,4)-BNDS(I,2)).LT.TOL	PROF 660
	1.AND.(BNDS(J,2)-BNDS(I,4)).LT.TOL.AND.(BNDS(J,4)-BNDS(I,4)).LT.TOL	PROF 662
	1)GO TO 8	PROF 664
	WRITE(6,103)ERROR(6),I,J	PROF 666
	IEXIT=1	PROF 668
	GO TO 8	PROF 670
C	-----	PROF 672
C	CHECK IF BOUNDARY (J) IS VERTICAL	PROF 674
C	-----	PROF 676
C	7 IF (ABS(DENOJ).GE.TOL)GO TO 11	PROF 678
	BI=(BNDS(I,2)-BNDS(I,4))/DENOI	PROF 680
	AI=BNDS(I,2)-BI*BNDS(I,1)	PROF 682
	Y1=AI+BI*BNDS(J,1)	PROF 684
C	-----	PROF 686
C	CHECK IF BOUNDARY (I) IS ABOVE VERTICAL BOUNDARY (J)	PROF 688
C	-----	PROF 690
C	IF((BNDS(J,2)-Y1).LT.TOL.AND.(BNDS(J,4)-Y1).LT.TOL)GO TO 8	PROF 692
	WRITE(6,103)ERROR(6),I,J	PROF 694
	IEXIT=1	PROF 696
	GO TO 8	PROF 698
C	9 BJ=(BNDS(J,2)-BNDS(J,4))/DENOJ	PROF 700
	AJ=BNDS(J,2)-BJ*BNDS(J,1)	PROF 702
	Y1=AJ+BJ*BNDS(I,1)	PROF 704
C	-----	PROF 706
C	-----	PROF 708
C	-----	PROF 710
C	-----	PROF 712
C	-----	PROF 714
C	-----	PROF 716
C	-----	PROF 718
C	-----	PROF 720
C	-----	PROF 722
C	-----	PROF 724
C	-----	PROF 726
C	-----	PROF 728
C	-----	PROF 730
C	-----	PROF 732

C	-----	PROF 734
C	CHECK IF VERTICAL BOUNDARY (I) IS ABOVE BOUNDARY (J)	PROF 736
C	-----	PROF 738
C	IF((Y1-BNDS(I,2)).LT.TOL.AND.(Y1-BNDS(I,4)).LT.TOL)GO TO 8	PROF 740
	WRITE(6,103)ERROR(6),I,J	PROF 742
	IEXIT=1	PROF 744
	GO TO 8	PROF 746
11	BI=(BNDS(I,2)-BNDS(I,4))/DENOI	PROF 748
	AI=BNDS(I,2)-BI*BNDS(I,1)	PROF 750
	BJ=(BNDS(J,2)-BNDS(J,4))/DENOJ	PROF 752
	AJ=BNDS(J,2)-BJ*BNDS(J,1)	PROF 754
	-----	PROF 756
C	CHECK IF BOUNDARIES (I) AND (J) ARE PARALLEL	PROF 758
C	-----	PROF 760
C	IF(ABS(BI-BJ).GE.TOL)GO TO 15	PROF 762
C	-----	PROF 764
C	CHECK IF BOUNDARY (I) IS ABOVE PARALLEL BOUNDARY (J)	PROF 766
C	-----	PROF 768
C	IF((AJ-AI).LT.TOL)GO TO 8	PROF 770
	WRITE(6,103)ERROR(6),I,J	PROF 772
	IEXIT=1	PROF 774
	GO TO 8	PROF 776
15	X1=(AI-AJ)/(BJ-BI)	PROF 778
	XP=X1+TOL	PROF 780
	XM=X1-TOL	PROF 782
	-----	PROF 784
C	CHECK IF BOUNDARY (I) IS ABOVE BOUNDARY (J) FOR REMAINING CASES	PROF 786
C	-----	PROF 788
C	IF(BNDS(J,1).GE.XM.AND.BI.GT.BJ)GO TO 8	PROF 790
	IF(BNDS(J,3).LE.XP.AND.BJ.GT.BI)GO TO 8	PROF 792
	IF(BNDS(I,1).GE.XM.AND.BI.GT.BJ)GO TO 8	PROF 794
	IF(BNDS(I,3).LE.XP.AND.BJ.GT.BI)GO TO 8	PROF 796
	WRITE(6,103)ERROR(6),I,J	PROF 798
	IEXIT=1	PROF 800
	8 CONTINUE	PROF 802
	-----	PROF 804
C	PRINT BOUNDARY DATA	PROF 806
C	-----	PROF 808
C	5 WRITE(6,108)(I,(BNDS(I,J),J=1,4),ITP(I),I=1,NBND)	PROF 810
108	FORMAT(12X,I3,3X,4F11.2,7X,I2)	PROF 812
	RETURN	PROF 814
	-----	PROF 816
C	*****	PROF 818
C	ENTRY SOIL	PROF 820
C	*****	PROF 822
C	ISOIL=1	PROF 824
	-----	PROF 826
C	PRINT SOIL TITLE	PROF 828
C	-----	PROF 830
C	WRITE(6,101)	PROF 832
101	FORMAT(1H1,9X,'ISOTROPIC SOIL PARAMETERS',//)	PROF 834
C		PROF 836
		PROF 838
		PROF 840
		PROF 842
		PROF 844
		PROF 846
		PROF 848
		PROF 850
		PROF 852
		PROF 854
		PROF 856
		PROF 858
		PROF 860

C	-----	PROF	862
C	READ NUMBER OF SOIL TYPES	PROF	864
C	-----	PROF	866
C	CALL READER(DUMMY,NSOI,0)	PROF	868
C	-----	PROF	870
C	-----	PROF	872
C	CHECK FOR SOIL TYPE PARAMETER CHANGES	PROF	874
C	-----	PROF	876
C	IF(NSOI.EQ.0)GO TO 18	PROF	878
C	NSOIL=NSOI	PROF	880
C	-----	PROF	882
C	PRINT NUMBER OF SOIL TYPES	PROF	884
C	-----	PROF	886
C	WRITE(6,109)NSOIL	PROF	888
C	109 FORMAT(10X,I2,*,TYPE(S) OF SOIL*)	PROF	890
C	-----	PROF	892
C	CHECK IF NUMBER OF SOIL TYPES IS GREATER	PROF	894
C	THAN OR EQUAL TO SOIL TYPE INDICES	PROF	896
C	-----	PROF	898
C	DO 17 I=1,NBND	PROF	900
C	IF(NSOIL.GE.1TP(I))GO TO 17	PROF	902
C	WRITE(6,104)ERROR(7),I	PROF	904
C	IEXIT=1	PROF	906
C	17 CONTINUE	PROF	908
C	-----	PROF	910
C	CHECK SOIL PARAMETER STORAGE LIMIT	PROF	912
C	-----	PROF	914
C	IF(NSOIL.LE.20)GO TO 21	PROF	916
C	WRITE(6,105)ERROR(8)	PROF	918
C	CALL QUIT	PROF	920
C	-----	PROF	922
C	PRINT SOIL PARAMETER HEADINGS	PROF	924
C	-----	PROF	926
C	21 WRITE(6,120)	PROF	928
C	120 FORMAT(//,	PROF	930
C	110X,*,SOIL*,3X,*,TOTAL*,3X,*,SATURATED*,3X,*,COHESION*,2X,	PROF	932
C	1*,*,FRICTION*,4X,*,PORE*,4X,*,PRESSURE*,2X,*,PIEZOMETRIC*,/,	PROF	934
C	110X,*,TYPE*,2X,*,UNIT WT.*,2X,*,UNIT WT.*,2X,*,INTERCEPT*,4X,	PROF	936
C	1*,*,ANGLE*,3X,*,PRESSURE*,2X,*,CONSTANT*,4X,*,SURFACE*,/,	PROF	938
C	111X,*,NO.*,3X,*,(PCF)*,5X,*,(PCF)*,6X,*,(PSF)*,6X,	PROF	940
C	1*,*,(DEG)*,2X,*,PARAMETER*,4X,*,(PSF)*,8X,*,NO.*,/,	PROF	942
C	-----	PROF	944
C	READ SOIL PARAMETERS	PROF	946
C	-----	PROF	948
C	DO 6 I=1,NSOIL	PROF	950
C	CALL READER(GAMMA(I),IDUMMY,1)	PROF	952
C	CALL READER(GSAT(I),IDUMMY,1)	PROF	954 -
C	CALL READER(C(I),IDUMMY,1)	PROF	956
C	CALL READER(PHI(I),IDUMMY,1)	PROF	958 -
C	CALL READER(RU(I),IDUMMY,1)	PROF	960
C	CALL READER(CU(I),IDUMMY,1)	PROF	962 -
C	CALL READER(DUMMY,NP(I),0)	PROF	964
C	6 CONTINUE	PROF	966
C		PROF	968
C		PROF	970
C		PROF	972
C		PROF	974
C		PROF	976
C		PROF	978
C		PROF	980
C		PROF	982
C		PROF	984
C		PROF	986
C		PROF	987 -
C		PROF	988
C		PROF	990


```

C -----
C PRINT SOIL PARAMETERS
C -----
C      WRITE(6,110)(I,GAMMA(I),GSAT(I),C(I),PHI(I),RU(I),CU(I),NP(I),
1      I=1,NSOIL)
110 FORMAT(10X,I3,F9.1,F10.1,F11.1,F10.1,F9.2,F11.1,I10)
      RETURN
C -----
C CHECK IF SOIL DATA DEFINED BEFORE CHANGING IT
C -----
C 18 IF(NSOIL.GT.0)GO TO 24
      WRITE(6,105)ERROR(9)
      IEXIT=1
C -----
C READ AND PRINT NUMBER OF SOIL TYPE CHANGES
C -----
C 24 CALL READER(DUMMY,NCHA,0)
      WRITE(6,121)NCHA
121 FORMAT(IH0,
1 9X,ISOTROPIC PARAMETERS OF,I3,SOIL TYPE(S) HAVE BEEN CHANGED,I/)
C -----
C CHECK NUMBER OF SOIL TYPE CHANGES
C -----
C IF(NCHA.LE.NSOIL)GO TO 12
      WRITE(6,105)ERROR(10)
      IEXIT=1
C -----
C CHECK IF NUMBER OF SOIL TYPE
C CHANGES EXCEEDS STORAGE LIMITS
C -----
C IF(NCHA.GT.20)CALL QUIT
12 DO 22 I=1,NCHA
C -----
C READ INDICE OF SOIL TYPE TO BE CHANGED
C -----
C CALL READER(DUMMY,K,0)
C -----
C PRINT INDICE OF SOIL TYPE TO BE CHANGED
C -----
C WRITE(6,114)K
114 FORMAT(/,
115X,CHANGE SOIL TYPE,I3)
C -----
C CHECK IF SOIL TYPE TO BE CHANGED IS DEFINED
C -----
C IF(K.LE.NSOIL)GO TO 19
      WRITE(6,105)ERROR(11)
      IEXIT=1
C -----

```

PROF 998
 PROF 1000
 PROF 1001
 PROF 1002
 PROF 1004
 PROF 1006
 PROF 1008
 PROF 1010
 PROF 1012
 PROF 1014
 PROF 1016
 PROF 1018
 PROF 1020
 PROF 1022
 PROF 1024
 PROF 1026
 PROF 1028
 PROF 1030
 PROF 1032
 PROF 1034
 PROF 1036
 PROF 1038
 PROF 1040
 PROF 1042
 PROF 1044
 PROF 1046
 PROF 1048
 PROF 1050
 PROF 1052
 PROF 1054
 PROF 1056
 PROF 1058
 PROF 1060
 PROF 1062
 PROF 1064
 PROF 1066
 PROF 1068
 PROF 1070
 PROF 1072
 PROF 1074
 PROF 1076
 PROF 1078
 PROF 1080
 PROF 1082
 PROF 1084
 PROF 1086
 PROF 1088
 PROF 1090
 PROF 1092
 PROF 1094
 PROF 1096
 PROF 1098
 PROF 1100
 PROF 1102
 PROF 1104
 PROF 1106
 PROF 1108
 PROF 1110
 PROF 1112
 PROF 1114
 PROF 1116
 PROF 1118

C	-----	PROF1120
C	CHECK IF SOIL TYPE INDICE EXCEEDS STORAGE LIMITS	PROF1122
C	-----	PROF1124
C	IF(K.GT.20)CALL QUIT	PROF1126
C		PROF1128
C	-----	PROF1130
C	READ CHANGES TO SOIL TYPE	PROF1132
C	-----	PROF1134
C		PROF1136
C		PROF1138
C	19 CALL READER(GAMMA(K),IDUMMY,1)	PROF1140
C	CALL READER(GSAT(K),IDUMMY,1)	PROF1142
C	CALL READER(C(K),IDUMMY,1)	PROF1144
C	CALL READER(PHI(K),IDUMMY,1)	PROF1146
C	CALL READER(RU(K),IDUMMY,1)	PROF1148
C	CALL READER(CU(K),IDUMMY,1)	PROF1150
C	CALL READER(DUMMY,NP(I),0)	PROF1151 -
C	22 CONTINUE	PROF1152
C	-----	PROF1154
C	PRINT SOIL PARAMETER HEADINGS	PROF1156
C	-----	PROF1158
C		PROF1160
C	WRITE(6,120)	PROF1162
C		PROF1164
C	-----	PROF1166
C	PRINT SOIL PARAMETERS	PROF1168
C	-----	PROF1170
C		PROF1172
C		PROF1174
C	WRITE(6,110)(I,GAMMA(I),GSAT(I),C(I),PHI(I),RU(I),CU(I),NP(I),	PROF1176 -
C	1 I=1,NSOIL)	PROF1177 -
C	RETURN	PROF1178
C	END	PROF1180

SUBROUTINE ANISO		ANIO	2
-----		ANIO	4
-----		ANIO	6
SUBROUTINE ANISO		ANIO	8
-----		ANIO	10
-----		ANIO	12
		ANIO	14
FUNCTIONS -		ANIO	16
		ANIO	18
READS THE NUMBER OF ANISOTROPIC SOIL TYPES.		ANIO	20
		ANIO	22
IF EQUAL TO ZERO, EXISTING ANISOTROPIC SOIL DATA IS SUPPRESSED		ANIO	24
OR REACTIVATED IF PREVIOUSLY SUPPRESSED.		ANIO	26
		ANIO	28
IF GREATER THAN ZERO, READS, CHECKS, STORES, AND PRINTS		ANIO	30
ANISOTROPIC SOIL DATA.		ANIO	32
		ANIO	34
-----		ANIO	36
		ANIO	38
DEFINITIONS --		ANIO	40
		ANIO	42
CSA	ARRAY CONTAINING VALUES OF MOHR-COULOMB COHESION FOR EACH SPECIFIED DIRECTION RANGE OF EACH ANISOTROPIC SOIL TYPE.	ANIO	44
		ANIO	46
		ANIO	48
		ANIO	50
DIREC	ARRAY CONTAINING VALUES DEFINING THE COUNTERCLOCKWISE LIMIT OF EACH DIRECTION RANGE OF EACH ANISOTROPIC SOIL TYPE.	ANIO	52
		ANIO	54
		ANIO	56
		ANIO	58
DUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN AN INTEGER IS TO BE READ.	ANIO	60
		ANIO	62
		ANIO	64
ERROR	VARIABLE REPRESENTING CURRENT ERROR CODE.	ANIO	66
		ANIO	68
I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	ANIO	70
		ANIO	72
IDUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN A REAL NUMBER IS TO BE READ.	ANIO	74
		ANIO	76
		ANIO	78
IEXIT	CONTROL CODE WHICH TERMINATES EXECUTION, IF DATA IS INCONSISTANT WITH THE REQUIREMENTS OF THE PROGRAM.	ANIO	80
		ANIO	82
		ANIO	84
ISTR	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM HANDLING THE ANISOTROPIC SOIL STRENGTH DATA.	ANIO	86
		ANIO	88
		ANIO	90
ITPA	ARRAY CONTAINING THE INDICES OF ANISOTROPIC SOIL TYPES.	ANIO	92
		ANIO	94
		ANIO	96
J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	ANIO	98
		ANIO	100
ND	DO LOOP TERMINATOR.	ANIO	102
		ANIO	104
NDIREC	ARRAY CONTAINING THE NUMBER OF DIRECTION RANGES FOR EACH ANISOTROPIC SOIL TYPE.	ANIO	106
		ANIO	108
		ANIO	110
NSA	NUMBER READ AS NUMBER OF SOIL TYPES HAVING ANISOTROPIC STRENGTH PROPERTIES.	ANIO	112
		ANIO	114
		ANIO	116
NSAL	NUMBER OF SOIL TYPES HAVING ANISOTROPIC STRENGTH PROPERTIES.	ANIO	118
		ANIO	120
		ANIO	122
NSOIL	NUMBER OF SOIL TYPES SPECIFIED.	ANIO	124
		ANIO	126
PHIA	ARRAY CONTAINING VALUES OF THE MOHR-COULOMB ANGLE OF INTERNAL FRICTION FOR EACH SPECIFIED DIRECTION RANGE OF EACH ANISOTROPIC SOIL TYPE.	ANIO	128
		ANIO	130
		ANIO	132
		ANIO	134


```

C      QUIT      SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE.      ANIO 136
C      TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE      ANIO 138
C      PROGRAM.      ANIO 140
C      RD      FACTOR FOR CONVERSION OF DEGREES TO RADIANS.      ANIO 142
C      NOU76EB0 --      NOU76EB0 --
C      READER      SUBROUTINE THAT READS INTEGER OR REAL DATA IN FREE      ANIO 144
C      FORM FORMAT.      ANIO 146
C      ANIO 148
C      -----      ANIO 150
C      -----      ANIO 152
C      ANIO 154
C      COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLT,IREAD,ISEARC,      ANIO 156
C      1 IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL      ANIO 158 -
C      COMMON /BLK02/BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND,      ANIO 160
C      1 NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20)      ANIO 162 -
C      COMMON /BLK07/CSA(10,10),DIREC(10,10),ITPA(10),NDIREC(10),NSAL,      ANIO 164 -
C      1 PHIA(10,10)      ANIO 166 -
C      DIMENSION ERROR(7)      ANIO 168
C      DATA ERROR/4HAI01,4HAI02,4HAI03,4HAI04,4HAI05,4HAI06,4HAI07/      ANIO 170
C      ANIO 172
C      -----      ANIO 174
C      READ NUMBER OF ANISOTROPIC SOIL TYPES      ANIO 176
C      -----      ANIO 178
C      ANIO 180
C      CALL READER(DUMMY,NSA,0)      ANIO 182
C      ANIO 184
C      -----      ANIO 186
C      CHECK FOR DATA SUPPRESSION      ANIO 188
C      -----      ANIO 190
C      ANIO 192
C      IF(NSA.NE.0)GO TO 4      ANIO 194
C      IF(ISTR.EQ.0)GO TO 10      ANIO 196
C      ISTR=0      ANIO 198
C      WRITE(6,102)      ANIO 200
C      102 FORMAT(///,      ANIO 202
C      110X,ANISOTROPIC STRENGTH DATA HAS BEEN SUPPRESSED#)      ANIO 204
C      RETURN      ANIO 206
C      ANIO 208
C      -----      ANIO 210
C      CHECK IF DATA DEFINED BEFORE      ANIO 212
C      SUPPRESSING OR REACTIVATING IT      ANIO 214
C      -----      ANIO 216
C      ANIO 218
C      10 IF(NSAL.GT.0)GO TO 14      ANIO 220
C      WRITE(6,101)ERROR(1)      ANIO 222
C      101 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,/)      ANIO 224
C      IEXIT=1      ANIO 226
C      ANIO 228
C      -----      ANIO 230
C      REACTIVATE SUPPRESSED DATA      ANIO 232
C      -----      ANIO 234
C      ANIO 236
C      14 ISTR=1      ANIO 238
C      WRITE(6,103)      ANIO 240
C      103 FORMAT(///,      ANIO 242
C      110X,SUPPRESSED ANISOTROPIC STRENGTH DATA HAS BEEN REACTIVATED#)      ANIO 244
C      RETURN      ANIO 246
C      4 NSAL=NSA      ANIO 248
C      ANIO 250
C      -----      ANIO 252
C      PRINT NUMBER OF ANISOTROPIC SOIL TYPES      ANIO 254
C      -----      ANIO 256
C      ANIO 258
C      WRITE(6,109)NSAL      ANIO 260
C      109 FORMAT(///,      ANIO 262
C      1 9X,ANISOTROPIC STRENGTH PARAMETERS#,/      ANIO 264
C      113X,12,SOIL TYPE(S)#)      ANIO 266

```


C		ANIO 268
C		ANIO 270
C	-----	ANIO 272
C	CHECK IF NUMBER OF ANISOTROPIC SOIL	ANIO 274
C	TYPES EXCEEDS NUMBER OF SOIL TYPES	ANIO 276
C	-----	ANIO 278
	IF(NSAL.LE.NSOIL)GO TO 5	ANIO 280
	WRITE(6,101)ERROR(2)	ANIO 282
	IEXIT=1	ANIO 284
C	-----	ANIO 286
C	CHECK ANISOTROPIC SOIL PARAMETER STORAGE LIMIT	ANIO 288
C	-----	ANIO 290
C	5 IF(NSAL.LE.10)GO TO 3	ANIO 292
	WRITE(6,101)ERROR(3)	ANIO 294
	CALL QUIT	ANIO 296
	3 DO 9 I=1,NSAL	ANIO 298
C	-----	ANIO 300
C	READ AND PRINT INDICE OF ANISOTROPIC SOIL TYPE	ANIO 302
C	-----	ANIO 304
	CALL READER(DUMMY,ITPA(I),0)	ANIO 306
	WRITE(6,113)ITPA(I)	ANIO 308
	113 FORMAT(/,	ANIO 310
	110X,SOIL TYPE,I3,IS ANISTROPIC)	ANIO 312
C	-----	ANIO 314
C	CHECK SOIL TYPE NUMBER OF ANISOTROPIC SOIL	ANIO 316
C	-----	ANIO 318
	IF(ITPA(I).LE.NSOIL)GO TO 8	ANIO 320
	WRITE(6,104)ERROR(4),I	ANIO 322
	104 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,5X,9HSOIL TYPE,I3)	ANIO 324
	IEXIT=1	ANIO 326
C	-----	ANIO 328
C	READ AND PRINT NUMBER OF DIRECTION	ANIO 330
C	RANGES FOR ANISOTROPIC SOIL TYPE	ANIO 332
C	-----	ANIO 334
	8 CALL READER(DUMMY,NDIREC(I),0)	ANIO 336
	ND=NDIREC(I)	ANIO 338
	WRITE(6,114)ND	ANIO 340
	114 FORMAT(/,	ANIO 342
	110X,NUMBER OF DIRECTION RANGES SPECIFIED =,I3)	ANIO 344
C	-----	ANIO 346
C	CHECK DIRECTION RANGE STORAGE LIMIT	ANIO 348
C	-----	ANIO 350
	IF(ND.GT.1.AND.ND.LE.10)GO TO 6	ANIO 352
	WRITE(6,104)ERROR(5),I	ANIO 354
	CALL QUIT	ANIO 356
C	-----	ANIO 358
C	PRINT HEADINGS FOR ANISOTROPIC STRENGTH DATA	ANIO 360
C	-----	ANIO 362
	6 WRITE(6,112)	ANIO 364
	112 FORMAT(/,	ANIO 366
	110X,DIRECTION,4X,COUNTERCLOCKWISE,5X,COHESION,5X,FRICTION,ANIO 368	
	112X,RANGE,7X,DIRECTION LIMIT,4X,INTERCEPT,7X,ANGLE,/,	ANIO 370
	113X,NO.,12X,(DEG),12X,(PSF),9X,(DEG),/,	ANIO 372
C		ANIO 374
		ANIO 376
		ANIO 378
		ANIO 380
		ANIO 382
		ANIO 384
		ANIO 386
		ANIO 388
		ANIO 390
		ANIO 392
		ANIO 394
		ANIO 396
		ANIO 398

C	-----	ANIO 400
C	READ ANISOTROPIC STRENGTH DATA	ANIO 402
C	-----	ANIO 404
C		ANIO 406
	DO 2 J=1,ND	ANIO 408
	CALL READER(DIREC(I,J),IDUMMY,1)	ANIO 410
	CALL READER(CSA(I,J),IDUMMY,1)	ANIO 412
	CALL READER(PHIA(I,J),IDUMMY,1)	ANIO 414
	2 CONTINUE	ANIO 416
C	-----	ANIO 420
C		ANIO 422
C	CHECK INCREASING ORDER OF COUNTERCLOCKWISE	ANIO 424
C	LIMIT OF EACH DIRECTION RANGE	ANIO 426
C	-----	ANIO 428
C		ANIO 430
	DO 13 J=2,ND	ANIO 432
	IF(DIREC(I,J-1).LE.DIREC(I,J))GO TO 13	ANIO 434 -
	WRITE(6,104)ERROR(6),I	ANIO 436
	IEXIT=1	ANIO 438
	GO TO 11	ANIO 440
	13 CONTINUE	ANIO 442
C	-----	ANIO 444
C		ANIO 446
C	CHECK IF ALL THE REQUIRED DIRECTIONS ARE DEFINED	ANIO 448
C	-----	ANIO 450
C		ANIO 452
	11 IF(DIREC(I,ND).GE.90.)GO TO 7	ANIO 454 -
	WRITE(6,104)ERROR(7),I	ANIO 456
	IEXIT=1	ANIO 458
C	-----	ANIO 460
C		ANIO 462
C	PRINT ANISOTROPIC STRENGTH DATA	ANIO 464
C	-----	ANIO 466
C		ANIO 468
	7 WRITE(6,111)(J,DIREC(I,J),CSA(I,J),PHIA(I,J),J=1,ND)	ANIO 469 -
111	FORMAT(10X,15,9X,F9.1,9X,F9.1,4X,F9.1)	ANIO 470
	DO 12 J=1,ND	ANIO 471 -
	DIREC(I,J)=DIREC(I,J)*RD	ANIO 472 -
	12 CONTINUE	ANIO 473 -
	9 CONTINUE	ANIO 474
	ISTR=1	ANIO 476
	RETURN	ANIO 478
	END	ANIO 480

SUBROUTINE WATER		WATR	2
-----		WATR	4
-----		WATR	6
SUBROUTINE WATER		WATR	8
-----		WATR	10
-----		WATR	12
		WATP	14
FUNCTIONS -		WATP	16
		WATP	18
READS THE NUMBER OF PIEZOMETRIC SURFACES DEFINED.		WATR	20
		WATR	22
IF EQUAL TO ZERO, EXISTING WATER SURFACE DATA IS		WATR	24
SUPPRESSED OR REACTIVATED IF PREVIOUSLY SUPPRESSED.		WATR	26
		WATR	28
IF GREATER THAN ZERO, READS, CHECKS, STORES, AND PRINTS THE		WATR	30
WATER SURFACE DATA.		WATR	32
		WATR	34
-----		WATR	36
		WATR	38
DEFINITIONS -		WATR	40
		WATR	42
DUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN	WATR	44
	AN INTEGER IS TO BE READ.	WATR	46
		WATR	48
ERROR	VARIABLE REPRESENTING CURRENT ERROR CODE.	WATR	50
		WATP	52
I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	WATR	54
		WATP	56
IDUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN	WATR	58
	A REAL NUMBER IS TO BE READ.	WATR	60
		WATR	62
IEXIT	CONTROL CODE WHICH TERMINATES EXECUTION, IF DATA IS	WATR	64
	INCONSISTANT WITH THE REQUIREMENTS OF THE PROGRAM.	WATR	66
		WATR	68
IWAT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	WATR	70
	HANDLING THE WATER SURFACE SPECIFIED BY SUBROUTINE	WATR	72
	WATER.	WATR	74
		WATR	76
J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	NOU76EBO	78
J2	VALUE OF J ROUNDED DOWN TO AN EVEN INTEGER.	NOU76EBO	80
		NOU76EBO	82
N	VARIABLE USED TO TEMPORARY STORE AN ARRAY VARIABLE	AUG76EBO	84
	FOR A DO STATEMENT.	AUG76EBO	86
		AUG76EBO	88
NPI	NUMBER READ AS NUMBER OF PIEZOMETRIC SURFACES	WATR	90
	DEFINED.	WATR	92
		WATR	94
NPIEZ	NUMBER OF POINTS DEFINING THE WATER SURFACE.	WATR	96
		WATR	98
NPZ	NUMBER OF PIEZOMETRIC SURFACES DEFINED.	JUL76EBO	100
		JUL76EBO	102
QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE,	WATR	104
	TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE	WATR	106
	PROGRAM.	WATR	108
		WATR	110
READER	SUBROUTINE THAT READS INTEGER OR REAL DATA IN FREE	WATR	112
	FORM FORMAT.	WATR	114
		WATR	116
TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	WATR	118
		WATR	120
UWAT	UNITWEIGHT OF WATER. IF 0. IS SPECIFIED 62.4 PCF	JUL76EBO	122
	IS ASSUMED.	JUL76EBO	124
		JUL76EBO	126
XPIEZ	ARRAY CONTAINING X COORDINATES OF POINTS DEFINING	WATR	128
	WATER SURFACE.	WATR	130
		WATR	132

C	YPIEZ	ARRAY CONTAINING Y COORDINATES OF POINTS DEFINING	WATR 112
C		WATER SURFACE.	WATR 114
C			WATR 116
C		-----	WATR 118
C		-----	WATR 120
C			WATR 122
	COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPL0T,IREAD,ISEARC,		WATR 124
	1 IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL		WATR 126 -
	COMMON /BLK03/UWAT,NPZ,NPIEZ(10),XPIEZ(10,40),YPIEZ(10,40)		WATR 128 -
	DIMENSION ERROR(5)		WATR 130
	DATA ERROR/4HWA01,4HWA02,4HWA03,4HWA04,4HWA05/		WATR 132
C			WATR 134
C		-----	WATR 136
C	READ NUMBER OF PIEZOMETRIC SURFACES TO BE SPECIFIED		WATR 138 -
C		-----	WATR 140
C			WATR 142
C	CALL READER(DUMMY,NPI,0)		WATR 144
C			WATR 146
C		-----	WATR 148
C	CHECK FOR WATER SURFACE DATA SUPPRESSION		WATR 150
C		-----	WATR 152
	IF(NPI.NE.0)GO TO 2		WATR 154
	IF(IWAT.EQ.0) GO TO 10		WATR 156
	IWAT=0		WATR 158
	WRITE(6,107)		WATR 160
	107 FORMAT(///,		WATR 162
	110X,=WATER SURFACE DATA HAS BEEN SUPPRESSED=)		WATR 164
	RETURN		WATR 166
C			WATR 168
C		-----	WATR 170
C	CHECK IF DATA DEFINED BEFORE		WATR 172
C	SUPPRESSING OR REACTIVATING IT		WATR 174
C		-----	WATR 176
	10 IF(NPZ.GT.0)GO TO 12		WATR 178
	WRITE(6,101)ERROR(1)		WATR 180
	101 FORMAT(/,10X,14H**** ERROR - ,A4.6H ****, /)		WATR 182 -
	IEXIT=1		WATR 184
C			WATR 186
C			WATR 188
C		-----	WATR 190
C	REACTIVATE SUPPRESSED DATA		WATR 192
C		-----	WATR 194
	12 IWAT=1		WATR 196
	WRITE(6,108)		WATR 198
	108 FORMAT(///,		WATR 200
	110X,=SUPPRESSED WATER SURFACE DATA HAS BEEN REACTIVATED=)		WATR 202
	RETURN		WATR 204
	2 NPZ=NPI		WATR 206
	WRITE(6,103)NPZ		WATR 208
	103 FORMAT(1H1,9X,I2,		WATR 209
	1 = PIEZOMETRIC SURFACE(S) HAVE BEEN SPECIFIED=)		WATR 210
C			WATR 211
C		-----	WATR 212
C	READ UNITWEIGHT OF WATER		WATR 213
C		-----	WATR 214
	CALL READER(UWAT,IDUMMY,1)		WATR 215
	IF(UWAT.EQ.0.) UWAT= 62.4		WATR 216
	WRITE(6,106)UWAT		WATR 217
	106 FORMAT(//,10X,=UNITWEIGHT OF WATER =,F7.2)		WATR 218
C			WATR 219
			WATR 220
			WATR 221
			WATR 222

C	-----	WATR 223
C	PRINT NUMBER OF POINTS DEFINING THE WATER SURFACE	WATP 224
C	-----	WATR 225
C		WATR 226
	DO 3 J=1,NPZ	WATR 227
	CALL READER(DUMMY,NPIEZ(J),0)	WATR 228
	J2=(J/2)*2	WATR 229
	IF(J.NE.1.AND.J2.NE.J) WRITE(6,109)	WATR 230
109	FORMAT(1H1)	WATR 231
	WRITE(6,104)J,NPIEZ(J)	WATR 232
104	FORMAT(///,10X,*,PIEZOMETRIC SURFACE NO.,I3,*, SPECIFIED BY*,I3,	WATR 233
	1 *, COORDINATE POINTS*)	WATR 234
C		WATR 235
C	-----	WATR 236
C	CHECK WATER SURFACE STORAGE LIMIT	WATR 237
C	-----	WATR 239
	IF(NPIEZ(J).LE.40)GO TO 11	WATR 240
	WRITE(6,101)ERROR(2)	WATR 242
	CALL QUIT	WATR 244
C		WATR 246
C	-----	WATR 248
C	CHECK IF DATA IS SUFFICIENT	WATR 250
C	-----	WATR 252
C		WATR 254
11	IF(NPIEZ(J).GT.1)GO TO 13	WATR 256
	WRITE(6,101)ERROR(3)	WATR 258
	IEXIT=1	WATP 260
C		WATR 262
C	-----	WATR 264
C	PRINT WATER SURFACE DATA HEADINGS	WATR 266
C	-----	WATR 268
C		WATP 270
	13 WRITE(6,110)	WATR 272
110	FORMAT(//,	WATR 274
	1 12X,*,POINT*,6X,*,X-WATER*,5X,*,Y-WATER*,/,	WATR 276
	1 13X,*,NO.,*,9X,*,(FT)*,8X,*,(FT)*,/,	WATR 278
	N=NPIEZ(J)	WATR 279
	DO 8 I=1,N	WATP 280
C		WATR 282
C	-----	WATP 284
C	READ WATER SURFACE DATA	WATR 286
C	-----	WATR 288
	CALL READER(XPIEZ(J,I),IDUMMY,1)	WATP 290
	CALL READER(YPIEZ(J,I),IDUMMY,1)	WATR 292
C		WATR 294
C	-----	WATR 296
C	CHECK FOR 1ST QUADRANT LOCATION	WATR 298
C	-----	WATR 300
	IF(XPIEZ(J,I).GT.-TOL)GO TO 7	WATR 302
	GO TO 9	WATR 304
	7 IF(YPIEZ(J,I).GT.-TOL)GO TO 6	WATR 306
	9 WRITE(6,102)ERROR(4),J,I	WATR 308
102	FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,5X,12HSURFACE NO. ,I2,	WATP 310
	1 6H POINT,I3,/,	WATR 312
	IEXIT=1	WATR 314
C		WATR 315
		WATR 316
		WATR 318

C	-----	WATR	320
C	CHECK LEFT - RIGHT ORDER	WATR	322
C	-----	WATR	324
C		WATR	326
	6 IF(I.EQ.1)GO TO 8	WATR	328
	IF(XPIEZ(J,I-1).LT.XPIEZ(J,I))GO TO 8	WATR	330 -
	WRITE(6,102)ERROR(5),J,I	WATR	332
	IEXIT=1	WATR	334
	8 CONTINUE	WATR	336
C		WATR	338
C	-----	WATR	340
C	PRINT FREE WATER SURFACE DATA	WATR	342
C	-----	WATR	344
	WRITE(6,105)(I,XPIEZ(J,I),YPIEZ(J,I),I=1,N)	WATR	346
105	FORMAT(12X,I3,2X,2F12.2)	WATR	348 -
	3 CONTINUE	WATR	350
	IWAT=1	WATR	351 -
	RETURN	WATR	352
	END	WATR	354
		WATR	356


```

C      NSUC      NUMBER READ AS NUMBER OF BOUNDARY LOADS SPECIFIED.      LOAD 126
C      NSURC     NUMBER OF BOUNDARY LOADS SPECIFIED.                      LOAD 128
C      NTOP      NUMBER OF GROUND SURFACE BOUNDARIES.                    LOAD 130
C      QUIT      SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE,          LOAD 132
C                TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE    LOAD 134
C                PROGRAM.                                                 LOAD 136
C      RD        FACTOR FOR CONVERSION OF DEGREES TO RADIANS.            LOAD 138
C      READER     SUBROUTINE THAT READS INTEGER OR REAL DATA IN FREE    LOAD 140
C                FORM FORMAT.                                             LOAD 142
C      SURC      ARRAY CONTAINING X COORDINATES OF THE END POINTS        LOAD 144
C                DEFINING THE EXTENT OF LOADING.                         LOAD 146
C      TOL       TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.    LOAD 148
C      YSURC     ARRAY CONTAINING THE CALCULATED Y COORDINATES OF THE    LOAD 150
C                ENDS OF THE BOUNDARY LOADS ON THE GROUND SURFACE.      LOAD 152
C      -----LOAD 154
C      -----LOAD 156
C      COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLT,IREAD,ISEARC,    LOAD 158
C      1      IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL                  LOAD 160
C      COMMON /BLK02/BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NEND,  LOAD 162
C      1      NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20)                  LOAD 164
C      COMMON /BLK04/DELTA(10),LOAD(10),NSURC,SURC(10,2),YSURC(10,2)    LOAD 166
C      DIMENSION ERROR(5)                                                LOAD 168
C      DATA ERROR/4HLD01,4HLD02,4HLD03,4HLD04,4HLD05/                LOAD 170
C      REAL LOAD                                                         LOAD 172
C      -----LOAD 174
C      -----LOAD 176
C      READ NUMBER OF SURCHARGE BOUNDARY LOADS                          LOAD 178
C      -----LOAD 180
C      CALL READER(DUMMY,NSUC,0)                                         LOAD 182
C      -----LOAD 184
C      CHECK FOR SURCHARGE DATA SUPPRESSION                            LOAD 186
C      -----LOAD 188
C      IF(NSUC.NE.0)GO TO 9                                              LOAD 190
C      IF(ISURC.EQ.0)GO TO 10                                            LOAD 192
C      ISURC=0                                                            LOAD 194
C      WRITE(6,108)                                                       LOAD 196
C      108 FORMAT(///,                                                  LOAD 198
C      110X,'SURCHARGE BOUNDARY LOAD DATA HAS BEEN SUPPRESSED')      LOAD 200
C      RETURN                                                            LOAD 202
C      -----LOAD 204
C      CHECK IF DATA DEFINED BEFORE                                     LOAD 206
C      SUPPRESSING OR REACTIVATING IT                                   LOAD 208
C      -----LOAD 210
C      10 IF(NSURC.GT.0)GO TO 12                                         LOAD 212
C      WRITE(6,101)ERROR(1)                                              LOAD 214
C      101 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,/)              LOAD 216
C      IEXIT=1                                                            LOAD 218
C      -----LOAD 220
C      -----LOAD 222
C      -----LOAD 224
C      -----LOAD 226
C      -----LOAD 228
C      -----LOAD 230
C      -----LOAD 232
C      -----LOAD 234
C      -----LOAD 236
C      -----LOAD 238
C      -----LOAD 240
C      -----LOAD 242
C      -----LOAD 244
C      -----LOAD 246
C      -----LOAD 248
C      -----LOAD 250

```


C	-----	LOAD 252
C	REACTIVATE SUPPRESSED DATA	LOAD 254
C	-----	LOAD 256
C	12 ISURC=1	LOAD 258
	WRITE(6,109)	LOAD 260
	109 FORMAT(///,	LOAD 262
	110X, #SUPPRESSED BOUNDARY LOAD DATA HAS BEEN REACTIVATED#)	LOAD 264
	RETURN	LOAD 266
	9 NSURC=NSUC	LOAD 268
C	-----	LOAD 270
C	PRINT NUMBER OF SURCHARGE BOUNDARY LOADS	LOAD 272
C	-----	LOAD 274
C	WRITE(6,103)NSURC	LOAD 276
C	103 FORMAT(1H1,	LOAD 278
	1 9X, #BOUNDARY LOAD(S) #, //,	LOAD 280
	1 13X, I2, # LOAD(S) SPECIFIED#)	LOAD 282
C	-----	LOAD 284
C	CHECK BOUNDARY LOAD STORAGE LIMIT	LOAD 286
C	-----	LOAD 288
C	IF(NSURC.LE.10)GO TO 11	LOAD 290
	WRITE(6,101)ERROR(2)	LOAD 292
	CALL QUIT	LOAD 294
C	-----	LOAD 296
C	PRINT LOAD DATA HEADINGS	LOAD 298
C	-----	LOAD 300
C	11 WRITE(6,102)	LOAD 302
C	102 FORMAT(//,	LOAD 304
	1 10X, #LOAD#, 8X, #X-LEFT#, 6X, #X-RIGHT#, 5X, #INTENSITY#, 5X,	LOAD 306
	1 #DEFLECTION#, /,	LOAD 308
	1 11X, #NO. #, 9X, # (FT) #, 9X, # (FT) #, 6X, # (LB/SQFT) #, 7X, # (DEG) #, //)	LOAD 310
	DO 8 I=1, NSURC	LOAD 312
C	-----	LOAD 314
C	READ SURCHARGE BOUNDARY LOADING DATA	LOAD 316
C	-----	LOAD 318
C	CALL READER(SURC(I,1), IDUMMY,1)	LOAD 320
	CALL READER(SURC(I,2), IDUMMY,1)	LOAD 322
	CALL READER(LOAD(I), IDUMMY,1)	LOAD 324
	CALL READER(DELTA(I), IDUMMY,1)	LOAD 326
C	-----	LOAD 328
C	CHECK FOR 1ST QUADRANT LOCATION	LOAD 330
C	-----	LOAD 332
C	IF(SURC(I,1).GT.-TOL)GO TO 5	LOAD 334
	GO TO 4	LOAD 336
	5 IF(SURC(I,2).GT.-TOL)GO TO 3	LOAD 338
	4 WRITE(6,105)ERROR(3), I	LOAD 340
	105 FORMAT(/, 10X, 14H**** ERROR - , A4, 6H ****, 5X, 4HLOAD, I3, /)	LOAD 342
	IEXIT=1	LOAD 344
C	-----	LOAD 346
C	CHECK FOR LEFT-RIGHT ORDER OF BOUNDARY LOAD END POINTS	LOAD 348
C	-----	LOAD 350
C	3 IF(SURC(I,2).GT.SURC(I,1))GO TO 14	LOAD 352
	WRITE(6,105)ERROR(4), I	LOAD 354
	IEXIT=1	LOAD 356
C	-----	LOAD 358
C	LOAD 360	LOAD 362
C	LOAD 364	LOAD 366
C	LOAD 368	LOAD 370
C	LOAD 372	LOAD 374
C	LOAD 376	LOAD 378
C	LOAD 380	LOAD 382
C	LOAD 384	

C	-----	LOAD 386
C	CHECK LEFT-RIGHT ORDER OF BOUNDARY LOADS AND LOAD OVERLAP	LOAD 388
C	-----	LOAD 390
C		LOAD 392
	14 IF(I.EQ.1)GO TO 8	LOAD 394
	IF(SURC(I-1,2).LE.SURC(I,1))GO TO 8	LOAD 396
	WRITE(6,105)ERROR(5),I	LOAD 398
	IEXIT=1	LOAD 400
	8 CONTINUE	LOAD 402
C	-----	LOAD 404
C	PRINT LOAD DATA	LOAD 406
C	-----	LOAD 408
C		LOAD 410
	WRITE(6,104)(I,SURC(I,1),SURC(I,2),LOAD(I),DELTA(I),I=1,NSURC)	LOAD 412
	104 FORMAT(11X,I2,2X,2F13.2,2F13.1)	LOAD 414
	WRITE(6,106)	LOAD 416
	106 FORMAT(//,9X,NOTE - INTENSITY IS SPECIFIED AS A UNIFORMLY,	LOAD 418
	1 ≠ DISTRIBUTED≠,/,	LOAD 420
	1 16X,≠FORCE ACTING ON A HORIZONTALLY PROJECTED SURFACE.≠)	LOAD 422
C		LOAD 424
C	-----	LOAD 426
C	CALCULATE Y COORDINATES OF BOUNDARY LOADS	LOAD 428
C	-----	LOAD 430
C		LOAD 432
C		LOAD 434
	JS=1	LOAD 436
	DO 15 I=1,NSURC	LOAD 438
	DO 16 J=JS,NTOP	LOAD 440
	JTN=J	LOAD 441 -
	CALL INTSC2(BNDS(J,1),BNDS(J,2),BNDS(J,3),BNDS(J,4),0.,0.,0.,0.,	LOAD 442
	1 SURC(I,1),YSURC(I,1),INTS)	LOAD 444
	IF(INTS.EQ.1)GO TO 17	LOAD 446
	16 CONTINUE	LOAD 448
	17 JS=JTN	LOAD 450 -
	DO 18 J=JS,NTOP	LOAD 452
	JTN=J	LOAD 453 -
	CALL INTSC2(BNDS(J,1),BNDS(J,2),BNDS(J,3),BNDS(J,4),0.,0.,0.,0.,	LOAD 454
	1 SURC(I,2),YSURC(I,2),INTS)	LOAD 456
	IF(INTS.EQ.1)GO TO 19	LOAD 458
	18 CONTINUE	LOAD 460
	19 JS=JTN	LOAD 462 -
	DELTA(I)=DELTA(I)*RD	LOAD 464
	15 CONTINUE	LOAD 466
	ISURC=1	LOAD 468
	RETURN	LOAD 470
	END	LOAD 472


```

C      SUBROUTINE EQUAKE                                EQKE  2
C      -----EQKE  4
C      SUBROUTINE EQUAKE                                EQKE  6
C      -----EQKE  8
C      SUBROUTINE EQUAKE                                EQKE 10
C      -----EQKE 12
C      FUNCTIONS -                                     EQKE 14
C      -----EQKE 16
C      READS AND PRINTS THE PSEUDO-STATIC EARTHQUAKE COEFFICIENTS EQKE 18
C      AND CAVITATION PRESSURE.                       EQKE 20
C      -----EQKE 22
C      -----EQKE 24
C      -----EQKE 26
C      DEFINITIONS -                                   EQKE 28
C      -----EQKE 30
C      CAUT      CAVITATION PRESSURE.                  EQKE 32
C      -----EQKE 34
C      IDUMMY    DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN EQKE 36
C      A REAL NUMBER IS TO BE READ.                   EQKE 38
C      -----EQKE 40
C      KCOEF     HORIZONTAL EARTHQUAKE COEFFICIENT.     EQKE 42
C      -----EQKE 44
C      READER    SUBROUTINE THAT READS INTEGER OR REAL DATA IN FREE EQKE 46
C      FORM FORMAT.                                    EQKE 48
C      -----EQKE 50
C      UKCOEF    VERTICAL EARTHQUAKE COEFFICIENT.      EQKE 52
C      -----EQKE 54
C      -----EQKE 56
C      -----EQKE 58
C      -----EQKE 60
C      -----EQKE 62
C      COMMON /BLK02/BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND, EQKE 64
C      1      NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20) EQKE 66 -
C      COMMON /BLK11/CAUT,KCOEF,UKCOEF EQKE 68
C      REAL KCOEF EQKE 70
C      -----EQKE 72
C      -----EQKE 74
C      READ AND PRINT EARTHQUAKE COEFFICIENTS EQKE 76
C      -----EQKE 78
C      -----EQKE 80
C      CALL READER(KCOEF,IDUMMY,1) EQKE 82
C      CALL READER(UKCOEF,IDUMMY,1) EQKE 84
C      WRITE(6,102)KCOEF,UKCOEF EQKE 86
C      102 FORMAT(///, EQKE 88
C      110X, 'A HORIZONTAL EARTHQUAKE LOADING COEFFICIENT',/, EQKE 90
C      110X, 'OF',F5.3, ' HAS BEEN ASSIGNED',/, EQKE 92
C      110X, 'A VERTICAL EARTHQUAKE LOADING COEFFICIENT',/, EQKE 94
C      110X, 'OF',F5.3, ' HAS BEEN ASSIGNED') EQKE 96
C      -----EQKE 98
C      -----EQKE 100
C      READ AND PRINT CAVITATION PRESSURE EQKE 102
C      -----EQKE 104
C      -----EQKE 106
C      CALL READER(CAUT,IDUMMY,1) EQKE 108
C      WRITE(6,101)CAUT EQKE 110
C      101 FORMAT(/, EQKE 112
C      110X, 'CAVITATION PRESSURE =',F7.1, ' PSF',) EQKE 114
C      RETURN EQKE 116
C      END EQKE 118

```


C C C C C C C C C C C C C C C C	SUBROUTINE LIMITS		LIMIT	2
	-----		LIMIT	4
	-----		LIMIT	6
	SUBROUTINE LIMITS		LIMIT	8
	-----		LIMIT	10
	-----		LIMIT	12
	FUNCTIONS -		LIMIT	14
			LIMIT	16
	READS THE NUMBER OF SURFACE GENERATION LIMIT BOUNDARIES.		LIMIT	18
			LIMIT	20
	IF EQUAL TO ZERO, EXISTING LIMIT DATA IS SUPPRESSED OR		LIMIT	22
	REACTIVATED IF PREVIOUSLY SUPPRESSED.		LIMIT	24
			LIMIT	26
	IF GREATER THAN ZERO, READS, CHECKS, STORES, AND PRINTS LIMIT		LIMIT	28
	DATA.		LIMIT	30
C C C C C C C C C C C C C C C C	-----		LIMIT	32
			LIMIT	34
	-----		LIMIT	36
	DEFINITIONS -		LIMIT	38
			LIMIT	40
			LIMIT	42
	DUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN	LIMIT	44
		AN INTEGER IS TO BE READ.	LIMIT	46
			LIMIT	48
	ERROR	VARIABLE REPRESENTING CURRENT ERROR CODE.	LIMIT	50
			LIMIT	52
	I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	LIMIT	54
			LIMIT	56
	IDUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN	LIMIT	58
		A REAL NUMBER IS TO BE READ.	LIMIT	60
C C C C C C C C C C C C C C C C			LIMIT	62
	IEXIT	CONTROL CODE WHICH TERMINATES EXECUTION, IF DATA IS	LIMIT	64
		INCONSISTANT WITH THE REQUIREMENTS OF THE PROGRAM.	LIMIT	66
			LIMIT	68
	ILIMIT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	LIMIT	70
		AFFECTED BY SEARCHING LIMITS ESTABLISHED BY	LIMIT	72
		SUBROUTINE LIMITS.	LIMIT	74
			LIMIT	76
	J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	LIMIT	78
			LIMIT	80
	LIMIT	ARRAY CONTAINING THE COORDINATES OF THE END POINTS	LIMIT	82
		DEFINING EACH BOUNDARY USED TO CONFINE THE EXTENT OF	LIMIT	84
		SEARCHING BY SUBROUTINE RANDOM.	LIMIT	86
			LIMIT	88
	NLIM	NUMBER READ AS NUMBER OF SURFACE GENERATION LIMIT	LIMIT	90
		BOUNDARIES SPECIFIED.	LIMIT	92
C C C C C C C C C C C C C C C C			LIMIT	94
	NLIMIT	NUMBER OF SURFACE GENERATION LIMIT BOUNDARIES.	LIMIT	96
			LIMIT	98
	NLMT	NUMBER OF SURFACE GENERATION LIMIT BOUNDARIES WHICH	LIMIT	100
		DEFLECT GENERATED SURFACES UPWARD.	LIMIT	102
			LIMIT	104
	QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE,	LIMIT	106
		TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE	LIMIT	108
		PROGRAM.	LIMIT	110
			LIMIT	112
	READER	SUBROUTINE THAT READS INTEGER OR REAL DATA IN FREE	LIMIT	114
		FORM FORMAT.	LIMIT	116
			LIMIT	118
	TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	LIMIT	120
			LIMIT	122
	-----		LIMIT	124
	-----		LIMIT	126
			LIMIT	128
	COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLOT,IREAD,ISEARC,		LIMIT	130
	1	IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL	LIMIT	132
	COMMON /BLK06/LIMIT(20,4),NLIMIT,NLMT		LIMIT	134
	DIMENSION ERROR(S)		LIMIT	136

	DATA ERROR/4HLM01,4HLM02,4HLM03,4HLM04,4HLM05/	LIMIT 138
	REAL LIMIT	LIMIT 140
C		LIMIT 142
C	-----	LIMIT 144
C	READ NUMBER OF LIMIT BOUNDARIES	LIMIT 146
C	-----	LIMIT 148
C		LIMIT 150
	CALL READER(DUMMY,NLIM,0)	LIMIT 152
C		LIMIT 154
C	-----	LIMIT 156
C	CHECK FOR LIMIT DATA SUPPRESSION	LIMIT 158
C	-----	LIMIT 160
C		LIMIT 162
	IF(NLIM.NE.0)GO TO 5	LIMIT 164
	IF(ILIMIT.EQ.0)GO TO 6	LIMIT 166
	ILIMIT=0	LIMIT 168
	WRITE(6,107)	LIMIT 170
107	FORMAT(///,	LIMIT 172
	110X,SEARCH LIMIT BOUNDARY DATA HAS BEEN SUPPRESSED#)	LIMIT 174
	RETURN	LIMIT 176
C		LIMIT 178
C	-----	LIMIT 180
C	CHECK IF DATA DEFINED BEFORE	LIMIT 182
C	SUPPRESSING OR REACTIVATING IT	LIMIT 184
C	-----	LIMIT 186
C		LIMIT 188
	6 IF(NLIMIT.GT.0)GO TO 10	LIMIT 190
	WRITE(6,104)ERROR(1)	LIMIT 192
104	FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,/)	LIMIT 194
	IEXIT=1	LIMIT 196
C		LIMIT 198
C	-----	LIMIT 200
C	REACTIVATE SUPPRESSED DATA	LIMIT 202
C	-----	LIMIT 204
C		LIMIT 206
	10 ILIMIT=1	LIMIT 208
	WRITE(6,108)	LIMIT 210
108	FORMAT(///,	LIMIT 212
	110X,SUPPRESSED SEARCH LIMIT BOUNDARY DATA HAS BEEN REACTIVATED#)	LIMIT 214
	RETURN	LIMIT 216
	5 NLIMIT=NLIM	LIMIT 218
C		LIMIT 220
C	-----	LIMIT 222
C	PRINT NUMBER OF LIMIT BOUNDARIES	LIMIT 224
C	-----	LIMIT 226
C		LIMIT 228
	WRITE(6,103)NLIMIT	LIMIT 230
103	FORMAT(1H1,9X,SEARCHING ROUTINE WILL BE LIMITED TO AN AREA#,	LIMIT 232
	1 # DEFINED BY#,I3,# BOUNDARIES#)	LIMIT 234
C		LIMIT 236
C	-----	LIMIT 238
C	CHECK SEARCHING BOUNDARY STORAGE LIMIT	LIMIT 240
C	-----	LIMIT 242
C		LIMIT 244
	IF(NLIMIT.LE.20)GO TO 11	LIMIT 246
	WRITE(6,104)ERROR(3)	LIMIT 248
	CALL QUIT	LIMIT 250
C		LIMIT 252
C	-----	LIMIT 254
C	READ NUMBER OF BOUNDARIES THAT DEFLECT UPWARDS	LIMIT 256
C	-----	LIMIT 258
C		LIMIT 260
	11 CALL READER(DUMMY,NLMT,0)	LIMIT 262
	WRITE(6,111)NLMT	LIMIT 264
111	FORMAT(10X,	LIMIT 266
	1 #OF WHICH THE FIRST#,I3,# BOUNDARIES WILL DEFLECT SURFACES#,	LIMIT 268
	1 # UPWARD#,//)	LIMIT 270
C		LIMIT 272

C	-----	LIMIT	274
C	CHECK NUMBER OF UPWARD DEFLECTING BOUNDARIES	LIMIT	276
C	-----	LIMIT	278
C	IF(NLMT.LE.NLIMIT)GO TO 7	LIMIT	280
	WRITE(6,104)ERROR(2)	LIMIT	282
	IEXIT=1	LIMIT	284
		LIMIT	286
C	-----	LIMIT	288
C	PRINT HEADINGS FOR LIMIT DATA	LIMIT	290
C	-----	LIMIT	292
C		LIMIT	294
	7 WRITE(6,105)	LIMIT	296
	105 FORMAT(10X, #BOUNDARY#, 5X, #X-LEFT#, 5X, #Y-LEFT#, 4X, #X-RIGHT#, 4X,	LIMIT	298
	1 #Y-RIGHT#, /,	LIMIT	300
	1 13X, #NO. #, 8X, #(FT) #, 7X, #(FT) #, 7X, #(FT) #, 7X, #(FT) #, /)	LIMIT	302
		LIMIT	304
C	-----	LIMIT	306
C	READ BOUNDARY DATA, CHECK FOR 1ST QUADRANT	LIMIT	308
C	PROFILE, AND CHECK BOUNDARY END POINT ORDER	LIMIT	310
C	-----	LIMIT	312
C		LIMIT	314
	DO 2 I=1,NLIMIT	LIMIT	316
	DO 3 J=1,4	LIMIT	318
	CALL READER(LIMIT(I,J),IDUMMY,I)	LIMIT	320
	IF(LIMIT(I,J).GT.-TOL)GO TO 3	LIMIT	322
	WRITE(6,102)ERROR(4),I	LIMIT	324
	102 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,5X,8HBOUNDARY,I3, /)	LIMIT	326
	IEXIT=1	LIMIT	328
	3 CONTINUE	LIMIT	330
	IF((LIMIT(I,1)-LIMIT(I,3)).LT.TOL)GO TO 2	LIMIT	332
	WRITE(6,102)ERROR(5),I	LIMIT	334
	IEXIT=1	LIMIT	336
	2 CONTINUE	LIMIT	338
		LIMIT	340
C	-----	LIMIT	342
C	PRINT LIMIT BOUNDARY DATA	LIMIT	344
C	-----	LIMIT	346
C		LIMIT	348
	WRITE(6,106)(I,(LIMIT(I,J),J=1,4),I=1,NLIMIT)	LIMIT	350
	106 FORMAT(12X,I3,3X,4F11.2)	LIMIT	352
	ILIMIT=1	LIMIT	354
	RETURN	LIMIT	356
	END	LIMIT	358
		LIMIT	360

	SUBROUTINE INTSCT(X1,Y1,X2,Y2,X3,Y3,X4,Y4,X,Y,INTS)	ISCT	2
	-----	ISCT	4
	-----	ISCT	6
	-----	ISCT	8
	SUBROUTINE INTSCT	ISCT	10
	-----	ISCT	12
	-----	ISCT	14
	FUNCTIONS -	ISCT	16
	CHECKS FOR INTERSECTION OF TWO LINE SEGMENTS AND CALCULATES ITS	ISCT	18
	COORDINATES.	ISCT	20
	ENTRY INTSC2 *****	ISCT	22
	CHECKS FOR INTERSECTION OF A LINE SEGMENT WITH A VERTICAL LINE	ISCT	24
	AND CALCULATES ITS COORDINATES.	ISCT	26
	-----	ISCT	28
	-----	ISCT	30
	-----	ISCT	32
	-----	ISCT	34
	-----	ISCT	36
	DEFINITIONS -	ISCT	38
	ABS STANDARD FUNCTION WHICH DETERMINES THE ABSOLUTE	ISCT	40
	VALUE OF A NUMBER.	ISCT	42
	A12 INTERCEPT OF LINE THROUGH POINTS 1 AND 2.	ISCT	44
	A34 INTERCEPT OF LINE THROUGH POINTS 3 AND 4.	ISCT	46
	B12 SLOPE OF LINE THROUGH POINTS 1 AND 2.	ISCT	48
	B34 SLOPE OF LINE THROUGH POINTS 3 AND 4.	ISCT	50
	INTS CONTROL CODE WHICH SIGNALS WHETHER AN INTERSECTION	ISCT	52
	HAS OCCURRED OR NOT.	ISCT	54
	TOL TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	ISCT	56
	X X COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.	ISCT	58
	X1 X COORDINATE OF FIRST POINT.	ISCT	60
	X2 X COORDINATE OF SECOND POINT.	ISCT	62
	X3 X COORDINATE OF THIRD POINT.	ISCT	64
	X4 X COORDINATE OF FOURTH POINT.	ISCT	66
	Y Y COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.	ISCT	68
	Y1 Y COORDINATE OF FIRST POINT.	ISCT	70
	Y2 Y COORDINATE OF SECOND POINT.	ISCT	72
	Y3 Y COORDINATE OF THIRD POINT.	ISCT	74
	Y4 Y COORDINATE OF FOURTH POINT.	ISCT	76
	-----	ISCT	78
	-----	ISCT	80
	-----	ISCT	82
	-----	ISCT	84
	-----	ISCT	86
	-----	ISCT	88
	-----	ISCT	90
	-----	ISCT	92
	-----	ISCT	94
	-----	ISCT	96
	-----	ISCT	98
	-----	ISCT	100
	-----	ISCT	102
	-----	ISCT	104
	-----	ISCT	106
	-----	ISCT	108
	-----	ISCT	110
	-----	ISCT	112
	-----	ISCT	114
	-----	ISCT	116
	-----	ISCT	118
	-----	ISCT	120
	-----	ISCT	122
	COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPL0T,IPEAD,ISEARC,	ISCT	124
	1 IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL	ISCT	126
	INTS=0	ISCT	128
	-----	ISCT	130


```

C ----- ISCT 132
C CHECK FOR VERTICAL LINE SEGMENTS ISCT 134
C ----- ISCT 136
C IF (ABS(X1-X2).LT.TOL.AND.ABS(X3-X4).LT.TOL) RETURN ISCT 138
C IF (ABS(X1-X2).LT.TOL) GO TO 5 ISCT 140
C IF (ABS(X3-X4).LT.TOL) GO TO 6 ISCT 144
C ISCT 146
C ----- ISCT 148
C CHECK FOR AN INTERSECTION OF TWO NON-VERTICAL LINE SEGMENTS ISCT 150
C ----- ISCT 152
C B12=(Y1-Y2)/(X1-X2) ISCT 154
C A12=Y1-B12*X1 ISCT 156
C B34=(Y3-Y4)/(X3-X4) ISCT 158
C A34=Y3-B34*X3 ISCT 160
C IF (ABS(B12-B34).LT.TOL) RETURN ISCT 162
C X=(A12-A34)/(B34-B12) ISCT 164
C Y=A12+B12*X ISCT 166
C IF ((X1-X).LT.TOL.AND.(X3-X).LT.TOL.AND.(X-X2).LT.TOL.AND.(X-X4).LT ISCT 170
C 1.TOL) INTS=1 ISCT 172
C RETURN ISCT 174
C ISCT 176
C ----- ISCT 178
C CHECK FOR AN INTERSECTION, IF THE FIRST LINE SEGMENT IS VERTICAL ISCT 180
C ----- ISCT 182
C 5 X=X1 ISCT 184
C B34=(Y3-Y4)/(X3-X4) ISCT 186
C A34=Y3-B34*X3 ISCT 188
C Y=A34+X*B34 ISCT 190
C IF (((Y-Y2).LT.TOL.AND.(Y1-Y).LT.TOL).OR.((Y-Y1).LT.TOL.AND.(Y2-Y). ISCT 192
C 1LT.TOL)) INTS=1 ISCT 194
C IF ((X3-X).GT.TOL.OR.(X-X4).GT.TOL) INTS=0 ISCT 196
C RETURN ISCT 198
C ISCT 200
C ----- ISCT 202
C CHECK FOR AN INTERSECTION, IF THE SECOND LINE SEGMENT IS VERTICAL ISCT 204
C ----- ISCT 206
C 6 X=X3 ISCT 208
C B12=(Y1-Y2)/(X1-X2) ISCT 210
C A12=Y1-B12*X1 ISCT 212
C Y=A12+X*B12 ISCT 214
C IF (((Y-Y4).LT.TOL.AND.(Y3-Y).LT.TOL).OR.((Y-Y3).LT.TOL.AND.(Y4-Y). ISCT 216
C 1LT.TOL)) INTS=1 ISCT 218
C IF ((X1-X).GT.TOL.OR.(X-X2).GT.TOL) INTS=0 ISCT 220
C RETURN ISCT 222
C ***** ISCT 224
C ENTRY INTSC2 ISCT 226
C ***** ISCT 227
C INTS=0 ISCT 228
C IF (ABS(X1-X2).LT.TOL) RETURN ISCT 229
C ISCT 230
C ----- ISCT 232
C CHECK FOR AN INTERSECTION BETWEEN A ISCT 234
C LINE SEGMENT AND A VERTICAL LINE ISCT 236
C ----- ISCT 238
C B12=(Y1-Y2)/(X1-X2) ISCT 240
C A12=Y1-B12*X1 ISCT 242
C Y=A12+X*B12 ISCT 244
C IF ((X1-X).LT.TOL.AND.(X-X2).LT.TOL) INTS=1 ISCT 246
C RETURN ISCT 248
C END ISCT 250
C ISCT 252
C ISCT 254
C ISCT 256

```


	SUBROUTINE SURFAC	SURF	2
	-----SURF		4
	-----SURF		6
	SUBROUTINE SURFAC	SURF	8
	-----SURF		10
	-----SURF		12
	SURF		14
FUNCTIONS -		SURF	16
		SURF	18
READS, CHECKS, STORES, AND PRINTS DATA FOR A SPECIFIED TRIAL		SURF	20
FAILURE SURFACE.		SURF	22
		SURF	24
-----SURF			26
DEFINITIONS -		SURF	28
		SURF	30
		SURF	32
A	INTERCEPT OF LINE DEFINING GROUND SURFACE WHERE IT IS	SURF	34
	INTERSECTED BY THE TRIAL FAILURE SURFACE.	SURF	36
		SURF	38
ABS	STANDARD FUNCTION WHICH DETERMINES THE ABSOLUTE	SURF	40
	VALUE OF A NUMBER.	SURF	42
		SURF	44
B	SLOPE OF LINE DEFINING GROUND SURFACE WHERE IT IS	SURF	46
	INTERSECTED BY THE TRIAL FAILURE SURFACE.	SURF	48
		SURF	50
BNDS	AN ARRAY CONTAINING THE COORDINATES OF THE END POINTS	SURF	52
	DEFINING THE GROUND SURFACE AND SUBSURFACE	SURF	54
	BOUNDARIES.	SURF	56
		SURF	58
DENOM	DENOMINATOR OF EXPRESSION CHECKED AS A PRECAUTION OF	SURF	60
	BEING ZERO.	SURF	62
		SURF	64
DUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN	SURF	66
	AN INTEGER IS TO BE READ.	SURF	68
		SURF	70
ERROR	VARIABLE REPRESENTING CURRENT ERROR CODE.	SURF	72
		SURF	74
I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	SURF	76
		SURF	78
IDUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN	SURF	80
	A REAL NUMBER IS TO BE READ.	SURF	82
		SURF	84
IEXIT	CONTROL CODE WHICH TERMINATES EXECUTION, IF DATA IS	SURF	86
	INCONSISTANT WITH THE REQUIREMENTS OF THE PROGRAM.	SURF	88
		SURF	90
ISURF	CONTROL CODE WHICH INDICATES DEFINITION OF SPECIFIED	SURF	92
	TRIAL FAILURE SURFACE.	SURF	94
		SURF	96
J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	SURF	98
		SURF	100
K	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	SURF	102
		SURF	104
NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	SURF	106
		SURF	108
NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	SURF	110
		SURF	112
QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE,	SURF	114
	TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE	SURF	116
	PROGRAM.	SURF	118
		SURF	120
READER	SUBROUTINE THAT READS INTEGER OR REAL DATA IN FREE	SURF	122
	FORM FORMAT.	SURF	124
		SURF	126
SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS	SURF	128
	DEFINING A TRIAL FAILURE SURFACE.	SURF	130
		SURF	132
TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	SURF	134
		SURF	136


```

C -----SURF 138
C -----SURF 140
C -----SURF 142
COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLDT,IREAD,ISEARC,SURF 144
1 IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL,SURF 146
COMMON /BLK02/ENDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND,SURF 148
1 NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20)SURF 150
COMMON /BLK05/NSURF,SURF(100,2)SURF 152
DIMENSION ERROR(6)SURF 154
DATA ERROR/4HSF01,4HSF02,4HSF03,4HSF04,4HSF05,4HSF06/,SURF 156
ISURF=1SURF 158
C -----SURF 160
C -----SURF 162
C READ AND PRINT NUMBER OFSURF 164
C TRIAL FAILURE SURFACE POINTS SURF 166
C -----SURF 168
C -----SURF 170
CALL READER(DUMMY,NSURF,0)SURF 172
WRITE(6,102)NSURFSURF 174
102 FORMAT(1H1,SURF 176
1 10X,=TRIAL FAILURE SURFACE SPECIFIED BY=,I3,SURF 178
1 = COORDINATE POINTS=)SURF 180
C -----SURF 182
C -----SURF 184
CHECK TRIAL FAILURE SURFACE STORAGE LIMITS SURF 186
C -----SURF 188
C -----SURF 190
IF(NSURF.LE.100)GO TO 9SURF 192
WRITE(6,106)ERROR(1)SURF 194
106 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,/)SURF 196
CALL QUITSURF 199
C -----SURF 200
C -----SURF 202
CHECK FOR SUFFICIENT DATA SURF 204
C -----SURF 206
C -----SURF 208
9 IF(NSURF.GT.1)GO TO 10SURF 210
WRITE(6,106)ERROR(2)SURF 212
IEXIT=1SURF 214
C -----SURF 216
C -----SURF 218
PRINT HEADINGS FOR SURFACE DATA SURF 220
C -----SURF 222
C -----SURF 224
10 WRITE(6,101)SURF 226
101 FORMAT(/,SURF 228
1 12X,=POINT=,6X,=X-SURF=,6X,=Y-SURF=,/,SURF 230
1 13X,=NO.=,8X,=(FT)=,8X,=(FT)=,/)SURF 232
DO 7 I=1,NSURFSURF 234
C -----SURF 236
C -----SURF 238
READ TRIAL FAILURE SURFACE DATA SURF 240
C -----SURF 242
C -----SURF 244
CALL READER(SURF(I,1),IDUMMY,1)SURF 246
CALL READER(SURF(I,2),IDUMMY,1)SURF 248
C -----SURF 250
C -----SURF 252
CHECK FOR 1ST QUADRANT LOCATION SURF 254
C -----SURF 256
C -----SURF 258
DO 6 J=1,2SURF 260
IF(SURF(I,J).GT.-TOL)GO TO 6SURF 262
WRITE(6,105)ERROR(3),ISURF 264
105 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,5X,5HPOINT,I3,/)SURF 266
IEXIT=1SURF 268
GO TO 8SURF 270
6 CONTINUE SURF 272
C -----SURF 274

```


C	-----	SURF	276
C	CHECK TRIAL FAILURE SURFACE POINT ORDER	SURF	278
C	-----	SURF	280
C		SURF	282
	8 IF(I.EQ.1)GO TO 7	SURF	284
	IF(SURF(I,1).GT.SURF(I-1,1))GO TO 7	SURF	286
	WRITE(6,105)ERROR(4),I	SURF	288
	IEXIT=1	SURF	290
	7 CONTINUE	SURF	292
C		SURF	294
C	-----	SURF	296
C	CHECK IF TRIAL FAILURE SURFACE ENTERS	SURF	298
C	WITHIN DEFINED EXTENT OF GROUND SURFACE	SURF	300
C	-----	SURF	302
C		SURF	304
	IF(SURF(1,1).GT.BNDS(NTOP,3))GO TO 4	SURF	306
	IF((BNDS(1,1)-SURF(1,1)).LT.TOL)GO TO 11	SURF	308
	4 WRITE(6,106)ERROR(5)	SURF	310
	IEXIT=1	SURF	312
	GO TO 13	SURF	314
C		SURF	316
C	-----	SURF	318
C	CALCULATE THE DOWNSLOPE INTERSECTION OF THE TRIAL	SURF	320
C	FAILURE SURFACE WITH THE GROUND SURFACE	SURF	322
C	-----	SURF	324
C		SURF	326
	11 DO 2 I=1,NTOP	SURF	328
	IF(BNDS(I,3).LT.SURF(1,1))GO TO 2	SURF	330
	DENOM=BNDS(I,1)-BNDS(I,3)	SURF	332
	IF(ABS(DENOM).LT.TOL)GO TO 13	SURF	334
	B=(BNDS(I,2)-BNDS(I,4))/DENOM	SURF	336
	A=BNDS(I,2)-B*BNDS(I,1)	SURF	338
	SURF(1,2)=A+B*SURF(1,1)	SURF	340
	GO TO 13	SURF	342
	2 CONTINUE	SURF	344
C		SURF	346
C	-----	SURF	348
C	CHECK IF TRIAL FAILURE SURFACE EXITS	SURF	350
C	WITHIN DEFINED EXTENT OF GROUND SURFACE	SURF	352
C	-----	SURF	354
C		SURF	356
	13 IF(SURF(NSURF,1).LT.BNDS(1,1))GO TO 5	SURF	358
	IF((SURF(NSURF,1)-BNDS(NTOP,3)).LT.TOL)GO TO 12	SURF	360
	5 WRITE(6,106)ERROR(6)	SURF	362
	IEXIT=1	SURF	364
	GO TO 14	SURF	366
C		SURF	368
C	-----	SURF	370
C	CALCULATE THE UPSLOPE INTERSECTION OF THE TRIAL	SURF	372
C	FAILURE SURFACE WITH THE GROUND SURFACE	SURF	374
C	-----	SURF	376
C		SURF	378
	12 DO 3 K=1,NTOP	SURF	380
	I=NTOP-K+1	SURF	382
	IF(BNDS(I,1).GT.SURF(NSURF,1))GO TO 3	SURF	384
	DENOM=BNDS(I,1)-BNDS(I,3)	SURF	386
	IF(ABS(DENOM).LT.TOL)GO TO 14	SURF	388
	B=(BNDS(I,2)-BNDS(I,4))/DENOM	SURF	390
	A=BNDS(I,2)-B*BNDS(I,1)	SURF	392
	SURF(NSURF,2)=A+B*SURF(NSURF,1)	SURF	394
	GO TO 14	SURF	396
	3 CONTINUE	SURF	398
C		SURF	400
C	-----	SURF	402
C	PRINT TRIAL FAILURE SURFACE DATA	SURF	404
C	-----	SURF	406
C		SURF	408
	14 WRITE(6,103)(I,SURF(I,1),SURF(I,2),I=1,NSURF)	SURF	410
	103 FORMAT(12X,I3,2X,2F12.2)	SURF	412
	RETURN	SURF	414
	END	SURF	416

C	SUBROUTINE RANDOM	RAND	2
C	-----	RAND	4
C		RAND	6
C	SUBROUTINE RANDOM	RAND	8
C	-----	RAND	10
C		RAND	12
C		RAND	14
C	FUNCTIONS -	RAND	16
C		RAND	18
C	READS IN DATA FOR SEARCHING PARAMETERS AND STORES THEM IN BLANKCOMMON.	RAND	20
C		RAND	22
C	CHECKS DATA FOR CONSISTENCY - STOPS EXECUTION IF NOT.	RAND	24
C		RAND	26
C	PRINTS OUT INPUT DATA.	RAND	28
C		RAND	30
C		RAND	32
C	CALLS SUBROUTINES PLTN, ENTRY PLT4, AND PLOTIN TO PLOT PROFILE	RAND	34
C	AND, IF SPECIFIED, THE PIEZOMETRIC SURFACES, SEARCHING LIMITS,	RAND	36
C	AND BOUNDARY LOADS.	RAND	38
C		RAND	40
C	SELECTS INITIATION POINTS AND DIRECTIONAL LIMITATIONS FOR TRIAL	RAND	42
C	SURFACE PROPAGATION.	RAND	44
C		RAND	46
C	CALLS SUBROUTINE RANSUF FOR GENERATION OF EACH IRREGULAR OR	RAND	48
C	CIRCULAR SURFACE.	RAND	50
C		RAND	52
C	CALLS SUBROUTINES BLKSUF OR BLOCK2 FOR GENERATION OF EACH	JUL76E80	
C	TRIAL FAILURE SURFACE OF SLIDING BLOCK SHAPE.	JUL76E80	
C		JUL76E80	
C	CALLS SUBROUTINES ENTRY PLT2 AND ENTRY PLOTN2 TO PLOT EACH	RAND	54
C	TRIAL SURFACE ON PROFILE.	RAND	55
C		RAND	56
C	CALLS SUBROUTINE EXECUT FOR CALCULATION OF THE FACTOR OF SAFETY	RAND	58
C	FOR EACH TRIAL SURFACE GENERATED.	RAND	60
C		RAND	62
C	ACCUMULATES AND PRINTS DATA FOR THE TEN MOST CRITICAL	RAND	64
C	TRIAL SURFACES.	RAND	65
C		RAND	66
C	CALLS SUBROUTINE PLOTIN TO REPEAT PLOT OF PROFILE AND, IF	RAND	68
C	SPECIFIED, PIEZOMETRIC SURFACES, SEARCHING LIMITS, AND	RAND	70
C	BOUNDARY LOADS.	RAND	71
C		RAND	72
C	CALLS SUBROUTINES ENTRY PLT3 AND ENTRY PLOTN3 TO PLOT THE	RAND	74
C	TEN MOST CRITICAL TRIAL SURFACES ON THE PROFILE.	RAND	76
C		RAND	80
C	-----	RAND	88
C		RAND	90
C	DEFINITIONS -	RAND	92
C		RAND	94
C	A INTERCEPT OF LINE DEFINING THE GROUND SURFACE WHERE	RAND	96
C	A TRIAL FAILURE SURFACE INITIATES.	RAND	98
C		RAND	100
C	ABS STANDARD FUNCTION WHICH DETERMINES THE ABSOLUTE	RAND	102
C	VALUE OF A NUMBER.	RAND	104
C		RAND	106
C	ANGS1 COUNTERCLOCKWISE DIRECTION LIMIT, WHICH MAY BE	RAND	108
C	MODIFIED DURING GENERATION OF SURFACES FROM A	RAND	110
C	PARTICULAR INITIATION POINT, AT WHICH INITIAL LINE	RAND	112
C	SEGMENT OF A TRIAL FAILURE SURFACE MAY PROJECT.	RAND	114
C		RAND	116
C	ANGS2 CLOCKWISE DIRECTION LIMIT, WHICH MAY BE MODIFIED	RAND	118
C	DURING GENERATION OF SURFACES FROM A PARTICULAR	RAND	120
C	INITIATION POINT, AT WHICH INITIAL LINE SEGMENT OF A	RAND	122
C	TRIAL FAILURE SURFACE MAY PROJECT.	RAND	124
C		RAND	126
C	ANG1 UNMODIFIED COUNTERCLOCKWISE DIRECTION LIMIT, FOR ALL	RAND	128
C	INITIATION POINTS, AT WHICH INITIAL LINE SEGMENT OF A	RAND	130
C	TRIAL FAILURE SURFACE MAY PROJECT.	RAND	132

C			RAND 134
C			RAND 136
C	ANG2	UNMODIFIED CLOCKWISE DIRECTION LIMIT, FOR ALL INITIATION POINTS, AT WHICH INITIAL LINE SEGMENT OF TRIAL FAILURE SURFACE MAY PROJECT.	RAND 138
C			RAND 140
C			RAND 142
C	ATAN	STANDARD FUNCTION THAT CALCULATES THE ARCTANGENT OF NUMBER.	RAND 144
C			RAND 146
C			RAND 148
C	B	SLOPE OF LINE DEFINING THE GROUND SURFACE WHERE A TRIAL FAILURE SURFACE INITIATES.	RAND 150
C			RAND 152
C			RAND 154
C	BGRID	SLOPE OF A SLIDING BLOCK BOX'S CENTERLINE.	RAND 156
C			RAND 158
C	BLKSUF	SUBROUTINE WHICH GENERATES INDIVIDUAL PSEUDO-RANDOM SURFACES OF SLIDING BLOCK SHAPE.	RAND 160
C			RAND 162
C			RAND 164
C	BLOCK2	SUBROUTINE WHICH GENERATES INDIVIDUAL PSEUDO-RANDOM SURFACES OF SLIDING BLOCK SHAPE WITH ACTIVE AND PASSIVE ZONES GENERATED ACCORDING TO THE RANKINE THEORY.	JUL76EBO
C			JUL76EBO
C			JUL76EBO
C			JUL76EBO
C	BNDS	AN ARRAY OF THE END COORDINATE POINTS DEFINING THE GROUND SURFACE AND SUBSURFACE BOUNDARIES.	RAND 166
C			RAND 168
C			RAND 170
C	BPT	POSITION ON GROUND SURFACE WHICH EACH TRIAL SURFACE MUST EXCEED WHEN IT TERMINATES.	RAND 172
C			RAND 174
C			RAND 176
C	DENOM	DENOMINATOR OF EXPRESSION CHECKED AS A PRECAUTION OF BEING ZERO.	RAND 178
C			RAND 180
C			RAND 182
C	DUMMY	DUMMY VARIABLE USED IN CALL OF SUBROUTINE READER WHEN AN INTEGER IS TO BE READ.	RAND 184
C			RAND 186
C			RAND 188
C	END	LAST POSITION ON THE GROUND SURFACE AT WHICH TRIAL SURFACES INITIATE.	RAND 190
C			RAND 192
C			RAND 194
C	EPT	POSITION ON GROUND SURFACE WHICH EACH TRIAL SURFACE MUST NOT EXCEED WHILE BEING GENERATED.	RAND 196
C			RAND 198
C			RAND 200
C	ERROR	VARIABLE REPRESENTING CURRENT ERROR CODE.	RAND 202
C			RAND 204
C	EXECUT	SUBROUTINE THAT DIRECTS CALCULATION OF THE FACTOR OF SAFETY FOR A SPECIFIED FAILURE SURFACE.	RAND 206
C			RAND 208
C			RAND 210
C	FLOAT	STANDARD FUNCTION THAT CONVERTS AN INTEGER NUMBER TO A REAL NUMBER.	RAND 212
C			RAND 214
C			RAND 216
C	FRTYFU	45 DEGREES IN RADIANS.	RAND 218
C			RAND 220
C	FS	FACTOR OF SAFETY.	RAND 222
C			RAND 224
C	FSS	ARRAY CONTAINING VALUES OF THE FACTOR OF SAFETY FOR THE TEN MOST CRITICAL TRIAL SURFACES OF THOSE GENERATED AND EXAMINED.	RAND 226
C			RAND 228
C			RAND 230
C			RAND 232
C	I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	RAND 234
C			RAND 236
C	IANGL	CONTROL CODE SIGNALS WHETHER TRIAL SURFACE INITIATION ANGLES HAVE BEEN SPECIFIED OR NOT.	RAND 238
C			RAND 240
C			RAND 242
C	IBLK	CONTROL CODE WHICH ACTIVATES GENERATION OF A SLIDING BLOCK SURFACE.	RAND 244
C			RAND 246
C			RAND 248
C	IBLK2	CONTROL CODE WHICH ACTIVATES GENERATION OF A SLIDING BLOCK SURFACE WITH RANKINE ACTIVE AND PASSIVE WEDGES.	JUL76EBO -
C			JUL76EBO -
C			JUL76EBO -
C	ICIRC	CONTROL CODE WHICH ACTIVATES GENERATION OF A CIRCULAR SURFACE.	RAND 250
C			RAND 252

NS	NUMBER OF POINTS DEFINING THE LAST SURFACE GENERATED.	RAND 360 RAND 362 RAND 364
NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	RAND 366 RAND 368
NSURFS	ARRAY CONTAINING VALUES OF THE NUMBER OF POINTS DEFINING EACH OF THE TEN MOST CRITICAL TRIAL SURFACES.	RAND 370 RAND 372 RAND 374 RAND 376
NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	RAND 378 RAND 380
NTRIA	TOTAL NUMBER OF TRIAL FAILURE SURFACES GENERATED.	RAND 382 RAND 384
NTRIAL	NUMBER OF TRIAL SURFACES GENERATED FROM EACH STARTING POINT.	RAND 386 RAND 388 RAND 390
NUMBER	EXTERNAL SUBROUTINE THAT ACTIVATES THE PLOTTING PEN TO PLOT NUMBERS.	NOU76EBO -- NOU76EBO -- NOU76EBO --
PERPEN	90 DEGREES IN RADIANS.	RAND 392 RAND 394
PLOTIN	SUBROUTINE THAT PLOTS WITH A PLOTTING DEVICE AXES WITH LABELS, PROFILE BOUNDARIES AND IF APPLICABLE, THE WATER SURFACE, SURFACE GENERATION BOUNDARIES, SURCHARGE BOUNDARY LOADS AND GENERATION BOXES FOR THE SLIDING BLOCK SEARCH.	RAND 396 RAND 398 RAND 400 RAND 402 RAND 404 RAND 406
PLOTN2	ENTRY OF SUBROUTINE PLOTIN WHICH PLOTS INDIVIDUAL TRIAL FAILURE SURFACES WITH A PLOTTING DEVICE.	RAND 408 RAND 410 RAND 412
PLOTN3	ENTRY OF SUBROUTINE PLOTIN WHICH PLOTS THE TEN MOST CRITICAL TRIAL FAILURE SURFACES WITH A PLOTTING DEVICE.	RAND 414 RAND 416 RAND 418 RAND 420
PLTN	SUBROUTINE THAT STORES THE AXES AND THEIR TICKMARK LABELS IN AN ARRAY TO BE LATER PRINTED ON THE LINE PRINTER.	RAND 422 RAND 424 RAND 426 RAND 428
PLT2	ENTRY OF SUBROUTINE PLTN WHICH STORES INDIVIDUAL TRIAL FAILURE SURFACES INTO THE ARRAY TO BE PRINTED.	RAND 430 RAND 432 RAND 434
PLT3	ENTRY OF SUBROUTINE PLTN WHICH STORES THE TEN MOST CRITICAL TRIAL FAILURE SURFACES IN THE ARRAY TO BE PRINTED.	RAND 436 RAND 438 RAND 440 RAND 442
PLT4	ENTRY OF SUBROUTINE PLTN WHICH STORES THE PROFILE GEOMETRY AND, IF APPLICABLE, THE WATER SURFACE, SURFACE GENERATION BOUNDARIES, SURCHARGE BOUNDARY LOADS AND A SPECIFIED TRIAL FAILURE SURFACE. PLT4 THEN PRINTS THE ENTIRE ARRAY AND ITS AXIS LABELS.	RAND 444 RAND 446 RAND 448 RAND 450 RAND 452 RAND 454
QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE, TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE PROGRAM.	RAND 456 RAND 458 RAND 460 RAND 462
RADS	5 DEGREES IN RADIANS.	RAND 464 RAND 466
RANSUF	SUBROUTINE WHICH GENERATES INDIVIDUAL PSEUDO-RANDOM SURFACES OF A CIRCULAR AND IRREGULAR NATURE.	RAND 468 RAND 470 RAND 472
RD	FACTOR FOR CONVERSION OF DEGREES TO RADIANS.	RAND 474 RAND 476
READER	SUBROUTINE THAT READS INTEGER OR REAL DATA IN FREE FORM FORMAT.	RAND 478 RAND 480 RAND 482
SCALER	SUBROUTINE WHICH DETERMINES THE SCALE TO FIT PROBLEM GEOMETRY WITHIN THE PLOT.	RAND 484 RAND 486

C			RAND 488
C	SLOPE	SLOPE IN RADIAN OF LINE DEFINING THE GROUND SURFACE WHERE A TRIAL SURFACE INITIATES.	RAND 490
C			RAND 492
C			RAND 494
C	SLP	SLOPE OF LINE EXTENDING FROM FIRST POINT ON GROUND SURFACE TO EITHER OF THE BOTTOM CORNERS OF THE FIRST BOX SPECIFIED FOR A SLIDING BLOCK SEARCH.	RAND 496
C			RAND 498
C			RAND 500
C			RAND 502
C	SORT	SUBROUTINE THAT SORTS TEN TRIAL FAILURE SURFACES BY MAGNITUDE OF FACTOR OF SAFETY.	RAND 504
C			RAND 506
C			RAND 508
C	START	FIRST POSITION ON THE GROUND SURFACE AT WHICH TRIAL SURFACES INITIATE.	RAND 510
C			RAND 512
C			RAND 514
C	SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS DEFINING A TRIAL FAILURE SURFACE.	RAND 516
C			RAND 518
C			RAND 520
C	SURFS	ARRAY CONTAINING SETS OF COORDINATE POINTS DEFINING EACH OF THE TEN MOST CRITICAL TRIAL SURFACES.	RAND 522
C			RAND 524
C			RAND 526
C	SYMBOL	EXTERNAL SUBROUTINE THAT ACTIVATES THE PLOTTING PEN TO PLOT CHARACTERS OR ON-CENTER SYMBOLS.	NOU76E80
C			NOU76E80
C			NOU76E80
C	TAL	INTEGER VARIABLE STORING THE NUMBER OF TRIAL SURFACES GENERATED IN A SEARCH, USED BY THE EXTERNAL SUBROUTINE NUMBER.	NOU76E80
C			NOU76E80
C			NOU76E80
C	TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	RAND 528
C			RAND 530
C	TSURF	LENGTH OF LINE SEGMENTS DEFINING TRIAL SURFACES.	RAND 532
C			RAND 534
C	WIDTH	ARRAY CONTAINING THE VALUES OF THE WIDTH OF EACH BOX.	RAND 536
C			RAND 538
C			RAND 540
C	XL	ARRAY CONTAINING VALUES OF THE X COORDINATE OF THE LEFT END OF EACH BOX CENTERLINE.	RAND 542
C			RAND 544
C			RAND 546
C	XR	ARRAY CONTAINING VALUES OF THE X COORDINATE OF THE RIGHT END OF EACH BOX CENTERLINE.	RAND 548
C			RAND 550
C			RAND 552
C	Y	Y COORDINATE OF POINT ON TOP EDGE OF SLIDING BLOCK BOX DIRECTLY BELOW A GROUND SURFACE BOUNDARY END POINT.	RAND 554
C			RAND 556
C			RAND 558
C			RAND 560
C	YBPT	Y COORDINATE OF LEFT TERMINATION LIMIT AT THE GROUND SURFACE.	RAND 562
C			RAND 564
C			RAND 566
C	YEPT	Y COORDINATE OF RIGHT TERMINATION LIMIT AT THE GROUND SURFACE.	RAND 568
C			RAND 570
C			RAND 572
C	YGL	Y COORDINATE OF POINT ON GROUND SURFACE DIRECTLY ABOVE THE LEFT END OF A SLIDING BLOCK BOX.	RAND 574
C			RAND 576
C			RAND 578
C	YGR	Y COORDINATE OF POINT ON GROUND SURFACE DIRECTLY ABOVE THE RIGHT END OF A SLIDING BLOCK BOX.	RAND 580
C			RAND 582
C			RAND 584
C	YL	ARRAY CONTAINING VALUES OF THE Y COORDINATE OF THE LEFT END OF EACH BOX CENTERLINE.	RAND 586
C			RAND 588
C			RAND 590
C	YLL	LEFT TOP CORNER OF A SLIDING BLOCK BOX.	RAND 592
C			RAND 594
C	YMIN	LOWEST DEPTH TO WHICH A TRIAL SURFACE MAY EXTEND.	RAND 596
C			RAND 598
C	YR	ARRAY CONTAINING VALUES OF THE Y COORDINATE OF THE RIGHT END OF EACH BOX CENTERLINE.	RAND 600
C			RAND 602
C			RAND 604
C	YRR	RIGHT TOP CORNER OF A SLIDING BLOCK BOX.	RAND 606
C			RAND 608


```

C -----RAND 610
C -----RAND 612
C -----RAND 614
COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLT,IREAD,ISEARC,RAND 616
1 IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL,RAND 618
COMMON /BLK02/BHDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND,RAND 620
1 NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20),RAND 622
COMMON /BLK05/NSURF,SURF(100,2),RAND 624
COMMON /BLK12/ANGS1,ANGS2,BPT,EPT,FRTYFU,FS,FSS(12),JJ,NSURFS(12),RAND 626
1 PERPEN,SURFS(100,2,12),TSURF,YBPT,YEPT,YMIN,RAND 628
COMMON /BLK13/NGRID,WIDTH(10),XL(10),XR(10),YL(10),YR(10),RAND 630
COMMON /BLK15/ M,MB,RAND 631 -
DIMENSION ERROR(20),RAND 632 - *
DATA ERROR/4HRC01,4HRC02,4HRC03,4HRC04,4HRC05,4HRC06,4HRC07,RAND 634
1 4HRC08,4HRC09,4HRC12,4HBK01,4HBK02,4HBK03,4HBK04,4HBK05,4HBK06,RAND 636
1 4HBK07,4HBK08,4HBK11,4HRC13/,RAND 638 - *
REAL INCRE,RAND 640
INTEGER TAL,RAND 641 -
PERPEN=90.*RD,RAND 642
RAD5=5.*RD,RAND 644
IANGL=0,RAND 646
IBLK=0,RAND 648
GO TO 19,RAND 650
C -----RAND 652
C -----RAND 654
C -----RAND 656
C -----RAND 658
C -----RAND 660
IBLK=1,RAND 662
NIPT=1,RAND 664
M=0,RAND 665 -
19 FRTYFU=45.*RD,RAND 666
ISEARC=1,RAND 668
ISURF=0,RAND 670
IF(ICIRC.EQ.0) MB=0,RAND 671 -
C -----RAND 672
C -----RAND 674
C -----RAND 676
C -----RAND 678
C -----RAND 680
IF(IBLK.EQ.1)GO TO 21,RAND 682
C -----RAND 684
C -----RAND 686
C -----RAND 688
C -----RAND 690
C -----RAND 692
IF(ICIRC.EQ.0)WRITE(6,101),RAND 694
101 FORMAT(1H1,9X),RAND 696
1≠A CRITICAL FAILURE SURFACE SEARCHING METHOD, USING A RANDOM ≠,RAND 698
1/,10X,RAND 700
1≠TECHNIQUE FOR GENERATING IRREGULAR SURFACES, HAS BEEN SPECIFIED.≠,RAND 702
1,/,),RAND 704
IF(ICIRC.EQ.1)WRITE(6,102),RAND 706
102 FORMAT(1H1,9X),RAND 708
1≠A CRITICAL FAILURE SURFACE SEARCHING METHOD, USING A RANDOM ≠,RAND 710
1/,10X,RAND 712
1≠TECHNIQUE FOR GENERATING CIRCULAR SURFACES, HAS BEEN SPECIFIED.≠,RAND 714
1,/,),RAND 716
C -----RAND 718
C -----RAND 720
C -----RAND 722
C -----RAND 724
C -----RAND 726
C -----RAND 728
CALL READER(DUMMY,NIPT,0),RAND 730
CALL READER(DUMMY,NTRIAL,0),RAND 732
C -----RAND 734

```

* See footnote p. 443.

C	-----	RAND 736
C	CALCULATE AND PRINT TOTAL NUMBER OF SURFACES GENERATED	RAND 738
C	-----	RAND 740
C	NTRIA=NTRIAL*NIPT	RAND 742
	WRITE(6,103)NTRIA	RAND 744
	103 FORMAT(8X,15, ' TRIAL SURFACES HAVE BEEN GENERATED.',//)	RAND 746
C	-----	RAND 748
C	CHECK FOR MINIMUM OF TEN SURFACES TO BE GENERATED	RAND 750
C	-----	RAND 752
C	IF(NTRIA.GE.10)GO TO 59	RAND 754
	WRITE(6,110)ERROR(10)	RAND 756
	IEXIT=1	RAND 758
C	-----	RAND 760
C	PRINT NUMBER OF INITIATION POINTS AND	RAND 762
C	NUMBER OF SURFACES GENERATED FROM EACH	RAND 764
C	-----	RAND 766
	59 WRITE(6,104)NTRIAL,NIPT	RAND 768
	104 FORMAT(8X,15,	RAND 770
	1 ' SURFACES INITIATE FROM EACH OF ',13, ' POINTS EQUALLY SPACED')	RAND 772
C	-----	RAND 774
C	READ AND PRINT LEFTMOST AND RIGHTMOST INITIATION POINTS	RAND 776
C	-----	RAND 778
	CALL READER(START,IDUMMY,1)	RAND 780
	CALL READER(END,IDUMMY,1)	RAND 782
	WRITE(6,118)START,END	RAND 784
	118 FORMAT(10X,	RAND 786
	1 'ALONG THE GROUND SURFACE BETWEEN X =',F7.2, ' FT.',/,	RAND 788
	139X, 'AND X =',F7.2, ' FT.',//)	RAND 790
C	-----	RAND 792
C	CHECK IF LEFTMOST INITIATION POINT DOES NOT	RAND 794
C	LIE BEYOND LEFT END OF DEFINED GROUND SURFACE	RAND 796
C	-----	RAND 798
	IF((BNDS(1,1)-START).LT.TOL)GO TO 6	RAND 800
	WRITE(6,110)ERROR(1)	RAND 802
	110 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,//)	RAND 804
	IEXIT=1	RAND 806
C	-----	RAND 808
C	CHECK ORDER OF INITIATION POINTS	RAND 810
C	-----	RAND 812
	6 IF((START-END).LT.TOL)GO TO 7	RAND 814
	WRITE(6,110)ERROR(2)	RAND 816
	IEXIT=1	RAND 818
C	-----	RAND 820
C	CHECK IF RIGHTMOST INITIATION POINT DOES NOT	RAND 822
C	LIE BEYOND RIGHT END OF DEFINED GROUND SURFACE	RAND 824
C	-----	RAND 826
	7 IF((END-BNDS(NTOP,3)).LT.TOL)GO TO 49	RAND 828
	WRITE(6,110)ERROR(3)	RAND 830
	IEXIT=1	RAND 832
C	-----	RAND 834
C	CHECK IF RIGHTMOST INITIATION POINT DOES NOT	RAND 836
C	LIE BEYOND RIGHT END OF DEFINED GROUND SURFACE	RAND 838
C	-----	RAND 840
	7 IF((END-BNDS(NTOP,3)).LT.TOL)GO TO 49	RAND 842
	WRITE(6,110)ERROR(3)	RAND 844
	IEXIT=1	RAND 846
C	-----	RAND 848
	7 IF((END-BNDS(NTOP,3)).LT.TOL)GO TO 49	RAND 850
	WRITE(6,110)ERROR(3)	RAND 852
	IEXIT=1	RAND 854
C	-----	RAND 856
	7 IF((END-BNDS(NTOP,3)).LT.TOL)GO TO 49	RAND 858
	WRITE(6,110)ERROR(3)	RAND 860
	IEXIT=1	RAND 860


```

C ----- RAND 862
C READ AND PRINT TERMINATION LIMITS RAND 864
C ----- RAND 866
C RAND 868
C 49 CALL READER(BPT,IDUMMY,1) RAND 870
C   CALL READER(EPT,IDUMMY,1) RAND 872
C   WRITE(6,105)BPT,EPT RAND 874
105 FORMAT(10X, 'EACH SURFACE TERMINATES BETWEEN X =#,F7.2,# FT.#,/, RAND 876
    138X, 'AND X =#,F7.2,# FT.#,/') RAND 878
C ----- RAND 880
C RAND 882
C CHECK IF RIGHT TERMINATION LIMIT LIES RAND 884
C WITHIN DEFINED GROUND SURFACE RAND 886
C ----- RAND 888
C RAND 890
C   IF((EPT-BNDS(NTOP,3)).LT.TOL)GO TO 9 RAND 892
C   WRITE(6,110)ERROR(4) RAND 894
C   IEXIT=1 RAND 896
C ----- RAND 898
C RAND 900
C CHECK ORDER OF TERMINATION LIMITS RAND 902
C ----- RAND 904
C RAND 906
C 9 IF((BPT-EPT).LT.TOL)GO TO 14 RAND 908
C   WRITE(6,110)ERROR(5) RAND 910
C   IEXIT=1 RAND 912
C ----- RAND 914
C RAND 916
C CHECK IF RIGHTMOST INITIATION POINT LIES RAND 918
C TO THE LEFT OF THE RIGHT TERMINATION LIMIT RAND 920
C ----- RAND 922
C RAND 924
C 14 IF((EPT-END).GT. 0.)GO TO 15 RAND 926
C   WRITE(6,110)ERROR(6) RAND 928
C   IEXIT=1 RAND 930
C ----- RAND 932
C RAND 934
C READ AND PRINT MINIMUM ELEVATION LIMITATION RAND 936
C ----- RAND 938
C RAND 940
C 15 CALL READER(YMIN,IDUMMY,1) RAND 942
C   WRITE(6,106)YMIN RAND 944
106 FORMAT(10X, RAND 946
    1 'UNLESS FURTHER LIMITATIONS WERE IMPOSED, THE MINIMUM ELEVATION#, RAND 948
    1/,10X, RAND 950
    1 'AT WHICH A SURFACE EXTENDS IS Y =#,F6.2,# FT.#,/') RAND 952
C ----- RAND 954
C RAND 956
C CHECK IF MINIMUM ELEVATION POSITIVE RAND 958
C ----- RAND 960
C RAND 962
C   IF(YMIN.GE.0.)GO TO 16 RAND 964
C   WRITE(6,110)ERROR(7) RAND 966
C   IEXIT=1 RAND 968
C ----- RAND 970
C RAND 972
C READ AND PRINT LENGTH OF TRIAL FAILURE SURFACE LINE SEGMENT RAND 974
C ----- RAND 976
C RAND 978
C 16 CALL READER(TSURF,IDUMMY,1) RAND 980
C   WRITE(6,107)TSURF RAND 982
107 FORMAT(8X,F7.2, RAND 984
    1 '# FT. LINE SEGMENTS DEFINE EACH TRIAL FAILURE SURFACE.#,/') RAND 986
C ----- RAND 988

```


C	-----	RAND 990
C	CHECK IF LENGTH OF LINE SEGMENT IS POSITIVE NON-ZERO	RAND 992
C	-----	RAND 994
C	IF(TSURF.GT.0.)GO TO 17	RAND 996
	WRITE(6,110)ERROR(8)	RAND 998
	IEXIT=1	RAND1000
C	-----	RAND1002
C	READ ANGLE LIMITATIONS FOR INITIAL LINE SEGMENT	RAND1004
C	-----	RAND1006
C	17 CALL READER(ANGS1,IDUMMY,1)	RAND1008
C	CALL READER(ANGS2,IDUMMY,1)	RAND1010
C	-----	RAND1012
C	CHECK FOR AUTOMATIC ANGLE LIMITATIONS	RAND1014
C	-----	RAND1016
C	IF(ANGS1.EQ.0..AND.ANGS2.EQ.0.)GO TO 22	RAND1018
C	IANG1=1	RAND1020
C	-----	RAND1022
C	PRINT SPECIFIED ANGLE LIMITATIONS	RAND1024
C	-----	RAND1026
C	WRITE(6,109)ANGS2,ANGS1	RAND1028 -
109	FORMAT(10X,	RAND1030
	1#RESTRICTIONS HAVE BEEN IMPOSED UPON THE ANGLE OF INITIATION.#,	RAND1032
	1/,10X,	RAND1034
	1#THE ANGLE HAS BEEN RESTRICTED BETWEEN THE ANGLES OF#,F6.1,	RAND1036
	1# AND#,F6.1,# DEG.#,//)	RAND1038
	ANG1=ANGS1*RD	RAND1040
	ANG2=ANGS2*RD	RAND1042
C	-----	RAND1044
C	CHECK ORDER OF ANGLE LIMITATIONS	RAND1046
C	-----	RAND1048
C	IF((ANGS2-ANGS1).LT.TOL) GO TO 22	RAND1050
	WRITE(6,110) ERROR(20)	RAND1052
	IEXIT=1	RAND1054
	GO TO 22	RAND1056
C	-----	RAND1058 *
C	PRINT TITLE FOR SLIDING BLOCK SURFACE	RAND1060
C	-----	RAND1062
C	21 WRITE(6,117)	RAND1064
117	FORMAT(1H1,9X,	RAND1066
	1#A CRITICAL FAILURE SURFACE SEARCHING METHOD, USING A RANDOM #,	RAND1068
	1/,10X,	RAND1070
	1#TECHNIQUE FOR GENERATING SLIDING BLOCK SURFACES, HAS BEEN #,/	RAND1072
	110X,#SPECIFIED.#,//)	RAND1080
	IF(1BLK2.EQ.1)WRITE(6,121)	RAND1082
121	FORMAT(10X,	RAND1084
	1#THE ACTIVE AND PASSIVE PORTIONS OF THE SLIDING SURFACES#,,	RAND1086
	1 10X,#ARE GENERATED ACCORDING TO THE RANKINE THEORY.#,//)	RAND1088
C		RAND1090
		RAND1092
		RAND1094
		RAND1096
		RAND1097
		RAND1098
		RAND1099
		RAND1100
		RAND1101
		RAND1102
		RAND1104

* This change was made in January 1978, hence is not included in the listing of changes in Appendix D (Table D.1). The change should eliminate this frequently occurring error in input data. Error RC13 is added to the list of error messages in the revised STABL User Manual.

C	-----	RAND1106
C	READ AND PRINT NUMBER OF TRIAL SURFACES GENERATED	RAND1108
C	-----	RAND1110
C	CALL READER(DUMMY,NTRIAL,0)	RAND1112
	WRITE(6,103)NTRIAL	RAND1114
C		RAND1116
C	-----	RAND1118
C	CHECK FOR MINIMUM OF TEN SURFACES TO BE GENERATED	RAND1120
C	-----	RAND1122
C		RAND1124
	IF(NTRIAL.GE.10) GO TO 58	RAND1126
	WRITE(6,110)ERROR(19)	RAND1128
	IEXIT=1	RAND1130
C		RAND1132
C	-----	RAND1134
C	READ AND PRINT NUMBER OF BOXES	RAND1136
C	-----	RAND1138
C		RAND1140
	58 CALL READER(DUMMY,NGRID,0)	RAND1142
	WRITE(6,119)NGRID	RAND1144
119	FORMAT(8X,I3,	RAND1146
	1# BOXES SPECIFIED FOR GENERATION OF CENTRAL BLOCK BASE#,,/)	RAND1148
C		RAND1150
C	-----	RAND1152
C	CHECK SLIDING BLOCK BOX STORAGE LIMIT	RAND1154
C	-----	RAND1156
C		RAND1158
	IF(NGRID.LE.10)GO TO 35	RAND1160
	WRITE(6,110)ERROR(11)	RAND1162
	CALL QUIT	RAND1164
C		RAND1166
C	-----	RAND1168
C	READ AND PRINT LENGTH OF TRIAL FAILURE SURFACE LINE SEGMENT	RAND1170
C	-----	RAND1172
C		RAND1174
	35 CALL READER(TSURF,IDUMMY,1)	RAND1176
	WRITE(6,131)TSURF	RAND1178
131	FORMAT(10X,	RAND1180
	1#LENGTH OF LINE SEGMENTS FOR ACTIVE AND PASSIVE PORTIONS OF#,,/,	RAND1182
	110X#SLIDING BLOCK IS#,F6.1,,/)	RAND1184
C		RAND1186
C	-----	RAND1188
C	CHECK IF LENGTH OF LINE SEGMENT IS POSITIVE NON-ZERO	RAND1190
C	-----	RAND1192
C		RAND1194
	IF(TSURF.GT.0.)GO TO 56	RAND1196
	WRITE(6,110)ERROR(12)	RAND1198
	IEXIT=1	RAND1200
56	DO 25 I=1,NGRID	RAND1202
C		RAND1204
C	-----	RAND1206
C	READ DATA DEFINING SLIDING BLOCK BOXES	RAND1208
C	-----	RAND1210
C		RAND1212
	CALL READER(XL(I),IDUMMY,1)	RAND1214
	CALL READER(YL(I),IDUMMY,1)	RAND1216
	CALL READER(XR(I),IDUMMY,1)	RAND1218
	CALL READER(YR(I),IDUMMY,1)	RAND1220
	CALL READER(WIDTH(I),IDUMMY,1)	RAND1222
25	CONTINUE	RAND1224
C		RAND1226
		RAND1228


```

C -----
C PRINT HEADINGS FOR BOX DATA
C -----
C WRITE(6,120)
120 FORMAT(10X, 'BOX', 8X, 'X-LEFT', 5X, 'Y-LEFT', 4X, 'X-RIGHT', 4X,
1 'Y-RIGHT', 5X, 'WIDTH', /,
110X, 'NO.', 9X, '(FT)', 7X, '(FT)', 7X, '(FT)', 7X, '(FT)', 8X, '(FT)', /)
C -----
C CHECK IF SPECIFIED CENTERLINE INTERSECTIONS
C OF EACH BOX ARE IN PROPER LEFT-RIGHT ORDER
C -----
C DO 39 I=1,NGRID
C IF((XL(I)-XR(I)).LT.TOL)GO TO 39
C WRITE(6,111)ERROR(13),I
111 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,5X,3HBOX,I3,/)
C IEXIT=1
39 CONTINUE
C IF(NGRID.EQ.1)GO TO 37
C -----
C CHECK IF BOXES SPECIFIED ARE IN LEFT-RIGHT ORDER
C -----
C DO 26 I=2,NGRID
C IF(XR(I-1).LT.XL(I))GO TO 26
C WRITE(6,111)ERROR(14),I
C IEXIT=1
26 CONTINUE
C -----
C CHECK IF EACH BOX IS TOTALLY DEFINED WITHIN THE 1ST QUADRANT
C -----
C 37 DO 38 I=1,NGRID
C YLL=YL(I)-ABS(WIDTH(I)/2.)
C YRR=YR(I)-ABS(WIDTH(I)/2.)
C IF(XL(I).GT.-TOL.AND.XR(I).GT.-TOL.AND.YLL.GT.-TOL.AND.YRR.GT.-TOL)
1 GO TO 38
C WRITE(6,111)ERROR(15),I
C IEXIT=1
38 CONTINUE
C -----
C CHECK IF EACH BOX IS TOTALLY BELOW THE DEFINED GROUND SURFACE
C -----
C JB=1
C DO 41 I=1,NGRID
C DO 42 J=JB,NTOP
C JTN=J
C CALL INTSC2(BNDS(J,1),BNDS(J,2),BNDS(J,3),BNDS(J,4),0.,0.,0.,0.,
1 XL(I),YGL,INTS)
C IF(INTS.EQ.1)GO TO 43
42 CONTINUE
43 JB=JTN
C YLL=YL(I)+WIDTH(I)/2.
C IF(YGL.GE.YLL)GO TO 44
C WRITE(6,111)ERROR(16),I
C IEXIT=1
C GO TO 41
44 DO 46 J=JB,NTOP
C JTN=J
C CALL INTSC2(BNDS(J,1),BNDS(J,2),BNDS(J,3),BNDS(J,4),0.,0.,0.,0.,
1 XR(I),YGR,INTS)
C IF(INTS.EQ.1)GO TO 47
46 CONTINUE

```

RAND1230
 RAND1232
 RAND1234
 RAND1236
 RAND1238
 RAND1240
 RAND1242
 RAND1244
 RAND1246
 RAND1248
 RAND1250
 RAND1252
 RAND1254
 RAND1256
 RAND1258
 RAND1260
 RAND1262
 RAND1264
 RAND1266
 RAND1268
 RAND1270
 RAND1272
 RAND1274
 RAND1276
 RAND1278
 RAND1280
 RAND1282
 RAND1284
 RAND1286
 RAND1288
 RAND1290
 RAND1292
 RAND1294
 RAND1296
 RAND1298
 RAND1300
 RAND1302
 RAND1304
 RAND1306
 RAND1308
 RAND1310
 RAND1312
 RAND1314
 RAND1316
 RAND1318
 RAND1320
 RAND1322
 RAND1324
 RAND1326
 RAND1328
 RAND1330
 RAND1332
 RAND1333 -
 RAND1334
 RAND1336
 RAND1338
 RAND1340
 RAND1342 -
 RAND1344
 RAND1346
 RAND1348
 RAND1350
 RAND1352
 RAND1354
 RAND1355 -
 RAND1356
 RAND1358
 RAND1360
 RAND1362


```

47 IF(JTN.EQ.JB)GO TO 48
   JB=JB+1
   BGRID=(YL(I)-YR(I))/(XL(I)-XR(I))
   DO 50 K=JB,JTN
   Y=YLL+BGRID*(BNDS(K,1)-XL(I))
   IF(BNDS(K,2).GE.Y)GO TO 50
   WRITE(6,111)ERROR(16),I
   IEXIT=1
   GO TO 41
50 CONTINUE
   JB=JTN
48 YRR=YR(I)+WIDTH(I)/2.
   IF(YGL.GE.YLL)GO TO 41
   WRITE(6,111)ERROR(16),I
   IEXIT=1
41 CONTINUE

-----
CHECK IF POSSIBLE TO COMPLETE TRIAL FAILURE
SURFACE FROM ANY POINT WITHIN LAST BOX
-----

   IF(XR(NGRID).LE.BNDS(NTOP,3))GO TO 53
   WRITE(6,110)ERROR(17)
   IEXIT=1

-----
CHECK IF POSSIBLE TO COMPLETE TRIAL FAILURE
SURFACE FROM ANY POINT WITHIN FIRST BOX
-----

53 DENOM=XL(1)-BNDS(1,1)
   IF(DENOM.LE.0.)GO TO 54
   SLP=(YL(1)-WIDTH(1)/2.-BNDS(1,2))/DENOM
   IF(SLP.GE.-1.0)GO TO 55
54 WRITE(6,110)ERROR(18)
   IEXIT=1
55 SLP=(YR(1)-WIDTH(1)/2.-BNDS(1,2))/(XR(1)-BNDS(1,1))
   IF(SLP.GE.-1.0)GO TO 36
   WRITE(6,110)ERROR(18)
   IEXIT=1

-----
PRINT DATA DEFINING SLIDING BLOCK BOXES
-----

36 WRITE(6,123)(I,XL(I),YL(I),XR(I),YR(I),WIDTH(I),I=1,NGRID)
123 FORMAT(9X,I3,4X,5F11.2)
   IF(1BLK2.EQ.1.AND.ILIMIT.NE.0)WRITE(6,112)
112 FORMAT(//,10X,
1 1#WARNING - LIMITATION BOUNDARIES HAVE BEEN SPECIFIED,#,/,
1 20X,#THESE ARE IGNORED IN THIS PROGRAM ROUTINE.#,/)
22 CALL SCALER
   CALL PLTN

-----
TERMINATE EXECUTION IF DATA FAULTY
-----

   IF(IEEXIT.EQ.0)GO TO 4
   CALL PLT4
   CALL QUIT
4 CALL PLOTIN
   II=0
   TAL=0
   ISORT=0

```

RAND1364 -
 RAND1365
 RAND1368
 RAND1370 -
 RAND1372
 RAND1374
 RAND1376
 RAND1378
 RAND1380
 RAND1382
 RAND1384 -
 RAND1386
 RAND1388
 RAND1390
 RAND1392
 RAND1394
 RAND1396
 RAND1398
 RAND1400
 RAND1402
 RAND1404
 RAND1406
 RAND1408
 RAND1410
 RAND1412
 RAND1414
 RAND1416
 RAND1418
 RAND1420
 RAND1422
 RAND1424
 RAND1426
 RAND1428
 RAND1430
 RAND1432 -
 RAND1434
 RAND1436
 RAND1438
 RAND1440 -
 RAND1442
 RAND1444
 RAND1446
 RAND1448
 RAND1450
 RAND1452
 RAND1453
 RAND1454
 RAND1455
 RAND1456
 RAND1457
 RAND1458
 RAND1459
 RAND1460
 RAND1462
 RAND1464
 RAND1466
 RAND1468
 RAND1470
 RAND1472
 RAND1474
 RAND1476
 RAND1478
 RAND1480
 RAND1482
 RAND1483 -
 RAND1484
 RAND1486


```

C -----
C CHECK FOR SLIDING BLOCK SURFACE SEARCH
C -----
C     IF (IBLK.EQ.1) GO TO 23
C     JJ=1
C -----
C CALCULATE HORIZONTAL SPACING BETWEEN INITIATION POINTS
C -----
C     INCRE=0
C     IF ((NIPT-1).NE.0) INCRE=(END-START)/FLOAT(NIPT-1)
C -----
C CHECK FOR CIRCULAR OR IRREGULAR SURFACE SEARCH
C -----
C     IF (ICIRC.EQ.0) GO TO 27
C -----
C CALCULATE Y COORDINATES OF TERMINATION
C LIMITS AT THE GROUND SURFACE
C -----
C     DO 28 I=1,NTOP
C     ITN=I
C     IF (BPT.LT.BNDS(I,1).OR.BPT.GE.BNDS(I,3)) GO TO 28
C     B=(BNDS(I,2)-BNDS(I,4))/(BNDS(I,1)-BNDS(I,3))
C     A=BNDS(I,2)-B*BNDS(I,1)
C     YBPT=B*BPT+A
C     GO TO 29
28 CONTINUE
29 JI=ITN
C     DO 30 I=JI,NTOP
C     IF (EPT.LE.BNDS(I,1).OR.EPT.GT.BNDS(I,3)) GO TO 30
C     B=(BNDS(I,2)-BNDS(I,4))/(BNDS(I,1)-BNDS(I,3))
C     A=BNDS(I,2)-B*BNDS(I,1)
C     YEPT=B*EPT+A
C     GO TO 27
30 CONTINUE
C -----
C SET UP INITIATION POINTS
C -----
C     27 N=0
C     START=START+TOL/2.
C     13 N=N+1
C -----
C DETERMINE COORDINATES OF INITIATION POINT
C -----
C     SURF(1,1)=START
C     START=START+INCRE
C     DO 5 I=JJ,NTOP
C     ITN=I

```

```

RAND1488
RAND1490
RAND1492
RAND1494
RAND1496
RAND1498
RAND1500
RAND1502
RAND1504
RAND1506
RAND1508
RAND1510
RAND1512
RAND1514
RAND1516
RAND1518
RAND1520
RAND1522
RAND1524
RAND1526
RAND1528
RAND1530
RAND1532
RAND1534
RAND1536
RAND1538
RAND1539
RAND1540
RAND1542
RAND1544
RAND1546
RAND1548
RAND1550
RAND1552
RAND1554
RAND1556
RAND1558
RAND1560
RAND1562
RAND1564
RAND1566
RAND1568
RAND1570
RAND1572
RAND1574
RAND1576
RAND1578
RAND1580
RAND1582
RAND1584
RAND1586
RAND1588
RAND1590
RAND1592
RAND1594
RAND1596
RAND1598
RAND1599
RAND1600

```



```

C -----
C CHECK IF INITIATION POINT LIES ON A
C VERTICAL GROUND SURFACE BOUNDARY
C -----
C   IF(BNDS(I,1).EQ.BNDS(I,3).AND.ABS(BNDS(I,1)-SURF(1,1)).LT.TOL)GO
1   TO 31
C   IF(SURF(1,1).LT.BNDS(I,1).OR.SURF(1,1).GE.BNDS(I,3))GO TO 5
C   B=(BNDS(I,2)-BNDS(I,4))/(BNDS(I,1)-BNDS(I,3))
C   A=BNDS(I,2)-B*BNDS(I,1)
C   SURF(1,2)=B*SURF(1,1)+A
C   IF(SURF(1,2).GE.YMIN)GO TO 52
C   WRITE(6,110)ERROR(9)
C   CALL QUIT
C -----
C DETERMINE DIRECTION LIMITS OF FIRST SURFACE SEGMENT
C -----
C 52 IF(IANGL.EQ.1)GO TO 12
C   SLOPE=ATAN(B)
C 33 ANG1=SLOPE-RADS
C   ANG2=-FRTYFU
C   GO TO 12
C -----
C DETERMINE Y COORDINATE OF AN INITIATION POINT AT
C THE BASE OF A VERTICAL GROUND SURFACE BOUNDARY
C -----
C 31 SURF(1,2)=BNDS(I,2)
C   SLOPE=PERPEN
C   IF(BNDS(I,4).GT.BNDS(I,2))GO TO 32
C   SURF(1,2)=BNDS(I,4)
C   DENOM=BNDS(I+1,1)-BNDS(I+1,3)
C   IF(DENOM.EQ.0.)GO TO 34
C   SLOPE=ATAN((BNDS(I+1,2)-BNDS(I+1,4))/DENOM)
C 32 IF(IANGL.EQ.1)GO TO 12
C   GO TO 33
C 34 IF(BNDS(I+1,2).LT.BNDS(I+1,4))GO TO 32
C   S CONTINUE
C 12 JJ=ITN
C   ANG1=ANG1
C   ANG2=ANG2
C -----
C GENERATE NTRIAL CIRCULAR OR IRREGULAR SURFACES FROM EACH
C INITIATION POINT OR NTRIAL TOTAL SLIDING BLOCK SURFACES
C -----
C 23 DO 2 I=1,NTRIAL
C   IF(IEEXIT.EQ.2) GO TO 18
C   TAL=TAL+1
C   IF(II.LT.11)II=II+1
C -----
C GENERATE AND PLOT SURFACE
C -----
C   IF(IBLK.EQ.0)CALL RANSUF
C   IF(IBLK.EQ.1.AND.IBLK2.EQ.0)CALL BLKSUF
C   IF(IBLK2.EQ.1)CALL BLOCK2
C   CALL PLOTN2
C   CALL PLT2

```

```

RAND1602
RAND1604
RAND1606
RAND1608
RAND1610
RAND1612
RAND1614
RAND1616
RAND1618
RAND1620
RAND1622
RAND1624
RAND1626
RAND1628
RAND1630
RAND1632
RAND1634
RAND1636
RAND1638
RAND1640
RAND1642
RAND1644
RAND1646
RAND1648
RAND1650
RAND1652
RAND1654
RAND1656
RAND1658
RAND1660
RAND1662
RAND1664
RAND1666
RAND1668
RAND1670
RAND1672
RAND1674
RAND1676
RAND1678
RAND1680
RAND1682
RAND1684
RAND1686
RAND1688
RAND1690
RAND1692
RAND1694
RAND1696
RAND1698
RAND1700
RAND1701
RAND1702
RAND1703
RAND1704
RAND1706
RAND1708
RAND1710
RAND1712
RAND1714
RAND1716
RAND1717
RAND1718
RAND1720
RAND1722
RAND1724

```



```

C -----
C CALCULATE FACTOR OF SAFETY
C -----
C      CALL EXECUT
C      FSS(11)=FS
C      NSURFS(11)=NSURF
C
C -----
C ACCUMULATE DATA FOR TEN CRITICAL SURFACES
C -----
C      DO 3 J=1,NSURF
C      SURFS(J,1,11)=SURF(J,1)
C      SURFS(J,2,11)=SURF(J,2)
3  CONTINUE
C      IF(11.LE.10)GO TO 2
C      IF(ISORT.EQ.1)GO TO 10
C
C -----
C SORT TEN SURFACES
C -----
C      CALL SORT
10 IF(FSS(11).GE.FSS(10))GO TO 11
C      NS=NSURFS(11)
C      DO 8 J=1,NS
C      SURFS(J,1,10)=SURFS(J,1,11)
C      SURFS(J,2,10)=SURFS(J,2,11)
8  CONTINUE
C      FSS(10)=FSS(11)
C      NSURFS(10)=NSURFS(11)
C      ISORT=0
C      GO TO 2
11 ISORT=1
2  CONTINUE
C      IF(1BLK.EQ.1)GO TO 24
C      IF(N.LT.NIPT)GO TO 13
24 IF(ISORT.EQ.1)GO TO 20
C
C -----
C SORT TEN SURFACES
C -----
C      CALL SORT
C
C -----
C PRINT COORDINATES OF POINTS DEFINING
C THE TEN CRITICAL TRIAL FAILURE SURFACES
C -----
C
18 IF(TAL.GE.10) GO TO 20
C      CALL PLT4
C      CALL QUIT
20 WRITE(6,108)
108 FORMAT(1H1,9X,/'FOLLOWING ARE DISPLAYED THE TEN MOST CRITICAL OF ',
1/'THE TRIAL',/,10X,/'FAILURE SURFACES EXAMINED. THEY ARE ORDERED',
1/' - MOST CRITICAL',/,10X,/'FIRST.',/))
C      IF(MB.EQ.1) WRITE(6,113)
113 FORMAT(10X,/'SAFETY FACTORS ARE CALCULATED BY THE MODIFIED BISHOP',
1/' METHOD.',/))
C      DO 57 K=1,10
C      K2=(K/2)*2
C      IF(K2.NE.K .AND. K.NE.1) WRITE(6,135)
135 FORMAT(1H1)

```

```

RAND1726
RAND1728
RAND1730
RAND1732
RAND1734
RAND1736
RAND1738
RAND1740
RAND1742
RAND1744
RAND1746
RAND1748
RAND1750
RAND1752
RAND1754
RAND1756
RAND1758
RAND1760
RAND1762
RAND1764
RAND1766
RAND1768
RAND1770
RAND1772
RAND1774
RAND1776
RAND1778
RAND1780
RAND1782
RAND1784
RAND1786
RAND1788
RAND1790
RAND1792
RAND1794
RAND1796
RAND1798
RAND1800
RAND1802
RAND1804
RAND1806
RAND1808
RAND1810
RAND1812
RAND1814
RAND1815
RAND1816
RAND1817
RAND1818
RAND1819
RAND1820
RAND1821
RAND1822
RAND1823
RAND1824
RAND1825
RAND1826
RAND1827
RAND1828
RAND1829
RAND1830
RAND1831
RAND1832
RAND1833
RAND1834

```


N=NSURFS(K)	RAND1836
WRITE(6,133)N	RAND1838
133 FORMAT(10X, #FAILURE SURFACE SPECIFIED BY#, I3,	RAND1840
1 # COORDINATE POINTS#, ///,	RAND1842
1 12X, #POINT#, 6X, #X-SURF#, 6X, #Y-SURF#, /	RAND1844
1 13X, #NO. #, 8X, # (FT) #, 8X, # (FT) #, /)	RAND1846
WRITE(6,132)(I, SURFS(I,1,K), SURFS(I,2,K), I=1,N)	RAND1848
132 FORMAT(12X, I3, 2X, 2F12.2)	RAND1850
WRITE(6,134)FSS(K)	RAND1852
134 FORMAT(//, 10X, #*** #, F15.3, # ***#, ///)	RAND1854 -
57 CONTINUE	RAND1856
CALL NUMBER (1.5,6.0,0.1,TAL,0.,2HI4)	RAND1859 -
CALL SYMBOL (2.0,6.0,0.1,28HSURFACES HAVE BEEN GENERATED,0.,28)	RAND1860 -
CALL PLT3	RAND1861
CALL PLT4	RAND1862
CALL PLOTIN	RAND1863
CALL PLOTN3	RAND1864
CALL SYMBOL (1.5,6.0,0.1,38H10 MOST CRITICAL OF SURFACES GENERATED	RAND1865
1,0.,38)	RAND1866
CALL SYMBOL (1.5,5.7,0.1,26HMINIMUM FACTOR OF SAFETY =,0.,26)	RAND1867
CALL NUMBER (3.8,5.7,0.1,FSS(1),0.,4HF7.3)	RAND1868
ISEARC=0	RAND1869
ICIRC=0	RAND1870
IBLK=0	RAND1871
IBLK2=0	RAND1872 -
MB=0	RAND1873 -
IEXIT=0	RAND1874 -
RETURN	RAND1875
END	RAND1876

SUBROUTINE RANSUF	RANS	2
-----	RANS	4
-----	RANS	6
SUBROUTINE RANSUF	RANS	8
-----	RANS	10
-----	RANS	12
-----	RANS	14
FUNCTIONS -	RANS	16
-----	RANS	18
GENERATES TRIAL FAILURE SURFACES OF CIRCULAR	RANS	20
OR IRREGULAR SHAPE BY A RANDOM TECHNIQUE.	RANS	22
-----	RANS	24
-----	RANS	26
-----	RANS	28
DEFINITIONS -	RANS	30
-----	RANS	32
A Y INTERCEPT OF THE PERPENDICULAR BISECTOR OF THE	RANS	34
INITIAL LINE SEGMENT GENERATED FOR A CIRCULAR	RANS	36
TRIAL FAILURE SURFACE.	RANS	38
-----	RANS	40
AB Y INTERCEPT OF THE PERPENDICULAR BISECTOR OF A LINE	RANS	42
SEGMENT WHOSE END POINTS ARE THE LEFT TERMINATION	RANS	44
LIMIT AT THE GROUND SURFACE AND THE RIGHT END POINT	RANS	46
OF THE INITIAL LINE SEGMENT OF A CIRCULAR TRIAL	RANS	48
FAILURE SURFACE.	RANS	50
-----	RANS	52
AE Y INTERCEPT OF THE PERPENDICULAR BISECTOR OF A LINE	RANS	54
SEGMENT WHOSE END POINTS ARE THE RIGHT TERMINATION	RANS	56
LIMIT AT THE GROUND SURFACE AND THE RIGHT END POINT	RANS	58
OF THE INITIAL LINE SEGMENT OF A CIRCULAR TRIAL	RANS	60
FAILURE SURFACE.	RANS	62
-----	RANS	64
ANGLMT THE MINIMUM INCLINATION ALLOWED FOR A GENERATED	RANS	66
TRIAL FAILURE SURFACE.	RANS	68
-----	RANS	70
ANGS1 COUNTERCLOCKWISE DIRECTION LIMIT, WHICH MAY BE	RANS	72
MODIFIED DURING GENERATION OF SURFACES FROM A	RANS	74
PARTICULAR INITIATION POINT, AT WHICH INITIAL LINE	RANS	76
SEGMENT OF A TRIAL FAILURE SURFACE MAY PROJECT.	RANS	78
-----	RANS	80
ANGS2 CLOCKWISE DIRECTION LIMIT, WHICH MAY BE MODIFIED	RANS	82
DURING GENERATION OF SURFACES FROM A PARTICULAR	RANS	84
INITIATION POINT, AT WHICH INITIAL LINE SEGMENT OF A	RANS	86
TRIAL FAILURE SURFACE MAY PROJECT.	RANS	88
-----	RANS	90
ATAN STANDARD FUNCTION THAT CALCULATES THE ARCTANGENT OF A	RANS	92
NUMBER.	RANS	94
-----	RANS	96
B SLOPE OF THE PERPENDICULAR BISECTOR OF THE INITIAL	RANS	98
LINE SEGMENT GENERATED FOR A CIRCULAR TRIAL FAILURE	RANS	100
SURFACE.	RANS	102
-----	RANS	104
BB SLOPE OF THE PERPENDICULAR BISECTOR OF A LINE SEGMENT	RANS	106
WHOSE END POINTS ARE THE LEFT TERMINATION LIMIT AT	RANS	108
THE GROUND SURFACE AND THE RIGHT END POINT OF THE	RANS	110
INITIAL LINE SEGMENT OF A CIRCULAR TRIAL FAILURE	RANS	112
SURFACE.	RANS	114
-----	RANS	116
BE SLOPE OF THE PERPENDICULAR BISECTOR OF A LINE SEGMENT	RANS	118
WHOSE END POINTS ARE THE RIGHT TERMINATION LIMIT AT	RANS	120
THE GROUND SURFACE AND THE RIGHT END POINT OF THE	RANS	122
INITIAL LINE SEGMENT OF A CIRCULAR TRIAL FAILURE	RANS	124
SURFACE.	RANS	126
-----	RANS	128
BNDS AN ARRAY OF THE END COORDINATE POINTS DEFINING THE	RANS	130
GROUND SURFACE AND SUBSURFACE BOUNDARIES.	RANS	132
-----	RANS	134

C	BPT	POSITION ON GROUND SURFACE WHICH EACH TRIAL SURFACE	RANS 136
C		MUST EXCEED WHEN IT TERMINATES.	RANS 138
C			RANS 140
C	COS	STANDARD FUNCTION THAT CALCULATES THE COSINE OF AN	RANS 142
C		ANGLE.	RANS 144
C			RANS 146
C	DTHETA	CHANGE IN DIRECTION BETWEEN TWO ADJACENT LINE	RANS 148
C		SEGMENTS OF A CIRCULAR TRIAL FAILURE SURFACE.	RANS 150
C			RANS 152
C	EPT	POSITION ON GROUND SURFACE WHICH EACH TRIAL SURFACE	RANS 154
C		MUST NOT EXCEED WHILE BEING GENERATED.	RANS 156
C			RANS 158
C	ERROR	VARIABLE REPRESENTING CURRENT ERROR CODE.	RANS 160
C			RANS 162
C	FRTYFU	45 DEGREES IN RADIANS.	RANS 164
C			RANS 166
C	I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	RANS 168
C			RANS 170
C	IANGL	CONTROL CODE SIGNALS WHETHER TRIAL SURFACE INITIATION	RANS 172
C		ANGLES HAVE BEEN SPECIFIED OR NOT.	RANS 174
C			RANS 176
C	ICIRC	CONTROL CODE WHICH ACTIVATES GENERATION OF A	RANS 184
C		CIRCULAR SURFACE.	RANS 186
C			RANS 188
C	II	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	RANS 190
C			RANS 192
C	ILIMIT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	RANS 194
C		AFFECTED BY SEARCHING LIMITS ESTABLISHED BY	RANS 196
C		SUBROUTINE LIMITS.	RANS 198
C			RANS 200
C	INTS	CONTROL CODE WHICH SIGNALS WHETHER AN INTERSECTION	RANS 202
C		HAS OCCURRED OR NOT.	RANS 204
C			RANS 206
C	INTSCT	SUBROUTINE WHICH DETERMINES WHETHER OR NOT TWO LINE	RANS 208
C		SEGMENTS INTERSECT AND CALCULATES THE COORDINATES OF	RANS 210
C		THE INTERSECTION.	RANS 212
C			RANS 214
C	ITN	VARIABLE USED TO TEMPORARY STORE THE VALUE OF AN	JUL76E80 -
C		INDEX VARIABLE USED FOR ARRAY SUBSCRIPTING.	JUL76E80 -
C			JUL76E80 -
C	J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	RANS 216
C			RANS 218
C	JJ	SUBSCRIPT OF LAST GROUND SURFACE BOUNDARY USED FOR	RANS 220
C		Y-COORDINATE CALCULATION OF THE INITIATION OF A TRIAL	RANS 222
C		SURFACE.	RANS 224
C			RANS 226
C	JT	SUBSCRIPT OF THE LAST GROUND SURFACE BOUNDARY USED TO	RANS 228
C		CHECK FOR AN INTERSECTION OF A TRIAL FAILURE SURFACE	RANS 230
C		WITH THE DEFINED GROUND SURFACE.	RANS 232
C			RANS 234
C	JTN	VARIABLE USED TO TEMPORARY STORE THE VALUE OF AN	JUL76E80 -
C		INDEX VARIABLE USED FOR ARRAY SUBSCRIPTING.	JUL76E80 -
C			JUL76E80 -
C	JTT	SUBSCRIPT OF THE LAST GROUND SURFACE BOUNDARY USED TO	RANS 236
C		CHECK FOR AN INTERSECTION OF A REDEFINED TRIAL	RANS 238
C		FAILURE SURFACE WITH THE GROUND SURFACE.	RANS 240
C			RANS 242
C	K	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	RANS 244
C			RANS 246
C	L	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	JUL76E80
C			JUL76E80 -
C	LIMIT	ARRAY CONTAINING THE COORDINATES OF THE END POINTS	RANS 248
C		DEFINING EACH BOUNDARY USED TO CONFINE THE EXTENT OF	RANS 250
C		SEARCHING BY BLOCK, RANDOM AND CIRCLE.	RANS 252
C			RANS 254
C	N	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	JUL76E80
C			JUL76E80

C	NLIMIT	NUMBER OF CONFINING BOUNDARIES SPECIFIED.	RANS 260
C			RANS 262 -
C	NLMT	NUMBER OF SURFACE GENERATION LIMIT BOUNDARIES WHICH	RANS 264
C		DEFLECT GENERATED SURFACES UPWARD.	RANS 266
C			RANS 268
C	NR	TOTAL NUMBER OF REJECTIONS WHILE ATTEMPTING TO	RANS 270
C		GENERATE A TRIAL FAILURE SURFACE.	RANS 272
C			RANS 274
C	NRR	NUMBER OF REJECTIONS WHILE ATTEMPTING TO GENERATE	RANS 276
C		A CIRCULAR SURFACE FROM THE SECOND LINE SEGMENT	JUL76EB0 -
C		WHEN CONFLICT OCCURES WITH RESTRICTIONS ON MINIMUM	JUL76EB0 -
C		ELEVATION AND OVERTURNING SLIP SURFACE.	JUL76EB0 -
C			RANS 282
C	NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	RANS 284
C			RANS 286
C	NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	RANS 288
C			RANS 290
C	PERPEN	90 DEGREES IN RADIANS.	RANS 292
C			RANS 294
C	PLT4	ENTRY OF SUBROUTINE PLTN WHICH STORES THE PROFILE	RANS 296
C		GEOMETRY AND, IF APPLICABLE, THE WATER SURFACE,	RANS 298
C		SURFACE GENERATION BOUNDARIES, SURCHARGE BOUNDARY	RANS 300
C		LOADS AND A SPECIFIED TRIAL FAILURE SURFACE. PLT4	RANS 302
C		THEN PRINTS THE ENTIRE ARRAY AND ITS AXIS LABELS.	RANS 304
C			RANS 306
C	QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE,	RANS 308
C		TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE	RANS 310
C		PROGRAM.	RANS 312
C			RANS 314
C	RANF	FUNCTION SUBPROGRAM THAT GENERATES A PSEUDO-RANDOM	RANS 320
C		NUMBER THAT HAS A UNIFORM PROBABILITY OF HAVING ANY	RANS 322
C		VALUE RANGING FROM ZERO TO ONE.	RANS 324
C			RANS 326
C	RB	RADIUS OF A CIRCLE CONTAINING THE LEFT TERMINATION	RANS 328
C		LIMIT AT THE GROUND SURFACE AND BOTH END POINTS OF	RANS 330
C		THE INITIAL LINE SEGMENT GENERATED FOR A CIRCULAR	RANS 332
C		TRIAL FAILURE SURFACE.	RANS 334
C			RANS 336
C	RBX	X COORDINATE OF THE CENTER OF A CIRCLE CONTAINING THE	RANS 338
C		LEFT TERMINATION LIMIT AT THE GROUND SURFACE AND	RANS 340
C		BOTH END POINTS OF THE INITIAL LINE SEGMENT GENERATED	RANS 342
C		FOR A CIRCULAR TRIAL FAILURE SURFACE.	RANS 344
C			RANS 346
C	RBY	Y COORDINATE OF THE CENTER OF A CIRCLE CONTAINING THE	RANS 348
C		LEFT TERMINATION LIMIT AT THE GROUND SURFACE AND	RANS 350 -
C		BOTH END POINTS OF THE INITIAL LINE SEGMENT GENERATED	RANS 352
C		FOR A CIRCULAR TRIAL FAILURE SURFACE.	RANS 354
C			RANS 356
C	RE	RADIUS OF A CIRCLE CONTAINING THE RIGHT TERMINATION	RANS 358
C		LIMIT AT THE GROUND SURFACE AND BOTH END POINTS OF	RANS 360
C		THE INITIAL LINE SEGMENT GENERATED FOR A CIRCULAR	RANS 362
C		TRIAL FAILURE SURFACE.	RANS 364
C			RANS 366
C	REX	X COORDINATE OF THE CENTER OF A CIRCLE CONTAINING THE	RANS 368
C		RIGHT TERMINATION LIMIT AT THE GROUND SURFACE AND	RANS 370 -
C		BOTH END POINTS OF THE INITIAL LINE SEGMENT GENERATED	RANS 372
C		FOR A CIRCULAR TRIAL FAILURE SURFACE.	RANS 374
C			RANS 376
C	REY	Y COORDINATE OF THE CENTER OF A CIRCLE CONTAINING THE	RANS 378
C		RIGHT TERMINATION LIMIT AT THE GROUND SURFACE AND	RANS 380
C		BOTH END POINTS OF THE INITIAL LINE SEGMENT GENERATED	RANS 382
C		FOR A CIRCULAR TRIAL FAILURE SURFACE.	RANS 384
C			RANS 386
C	SIN	STANDARD FUNCTION THAT CALCULATES THE SINE OF AN	RANS 388
C		ANGLE.	RANS 390
C			RANS 392

SLPB	SLOPE OF A LINE SEGMENT WHOSE END POINTS ARE THE LEFT TERMINATION LIMIT AT THE GROUND SURFACE AND THE RIGHT END POINT OF THE INITIAL LINE SEGMENT OF A CIRCULAR TRIAL FAILURE SURFACE.	RANS 394 RANS 396 RANS 398 RANS 400 RANS 402
SLPE	SLOPE OF A LINE SEGMENT WHOSE END POINTS ARE THE RIGHT TERMINATION LIMIT AT THE GROUND SURFACE AND THE RIGHT END POINT OF THE INITIAL LINE SEGMENT OF A CIRCULAR TRIAL FAILURE SURFACE.	RANS 404 RANS 406 RANS 408 RANS 410 RANS 412
SQRT	STANDARD FUNCTION FOR CALCULATION OF THE SQUARE ROOT OF A NUMBER.	RANS 414 RANS 416 RANS 418
SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS DEFINING A TRIAL FAILURE SURFACE.	RANS 420 RANS 422 RANS 424
TAN	STANDARD FUNCTION THAT CALCULATES TANGENS OF AN ANGLE.	RANS 432 RANS 434 RANS 436
THET	ANGLE AT WHICH A LINE SEGMENT OF A CIRCULAR SURFACE INTERSECTED A SURFACE GENERATION BOUNDARY.	RANS 438 RANS 440 RANS 442
THETA	DIRECTION OF LAST LINE SEGMENT DEFINING THE TRIAL SURFACE.	RANS 444 RANS 446 RANS 448
THETAB	MAXIMUM CHANGE IN DIRECTION ALLOWED BETWEEN ADJACENT LINE SEGMENTS OF A CIRCULAR TRIAL FAILURE SURFACE WITH A PARTICULAR INITIAL LINE SEGMENT.	RANS 450 RANS 452 RANS 454 RANS 456
THETA E	MINIMUM CHANGE IN DIRECTION ALLOWED BETWEEN ADJACENT LINE SEGMENTS OF A CIRCULAR TRIAL FAILURE SURFACE WITH A PARTICULAR INITIAL LINE SEGMENT.	RANS 458 RANS 460 RANS 462 RANS 464
THETAL	MINIMUM INCLINATION ALLOWED FOR A PARTICULAR LINE SEGMENT OF AN IRREGULAR TRIAL FAILURE SURFACE.	RANS 466 RANS 468 RANS 470
THETAS	INCLINATION OF THE INITIAL LINE SEGMENT OF A TRIAL FAILURE SURFACE.	RANS 472 RANS 474 RANS 476
THETAU	MAXIMUM INCLINATION ALLOWED FOR A PARTICULAR LINE SEGMENT OF AN IRREGULAR TRIAL FAILURE SURFACE.	RANS 478 RANS 480 RANS 482
THETA1	ANGLE SPREAD WITHIN WHICH THE NEXT LINE SEGMENT OF THE GENERATING TRIAL SURFACE MAY PROJECT.	RANS 484 RANS 486 RANS 488
THETA2	LIMITING ANGLE IN CLOCKWISE DIRECTION AT WHICH NEXT LINE SEGMENT MAY PROJECT.	RANS 490 RANS 492 RANS 494
TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	RANS 496 RANS 498
TSURF	LENGTH OF LINE SEGMENTS DEFINING TRIAL SURFACES.	RANS 500 RANS 502
TT	TANGENT OF THE INCLINATION OF THE INITIAL LINE SEGMENT OF A TRIAL FAILURE SURFACE.	RANS 504 RANS 506 RANS 508
X	X COORDINATE OF THE MIDPOINT OF THE INITIAL LINE SEGMENT OF A TRIAL FAILURE SURFACE.	RANS 516 RANS 518 RANS 520
XB	X COORDINATE OF THE MIDPOINT OF A LINE SEGMENT WHOSE END POINTS ARE THE LEFT TERMINATION LIMIT AT THE GROUND SURFACE AND THE RIGHT END POINT OF THE INITIAL LINE SEGMENT OF A CIRCULAR TRIAL FAILURE SURFACE.	RANS 522 RANS 524 RANS 526 RANS 528 RANS 530
XE	X COORDINATE OF THE MIDPOINT OF A LINE SEGMENT WHOSE END POINTS ARE THE RIGHT TERMINATION LIMIT AT THE GROUND SURFACE AND THE RIGHT END POINT OF THE INITIAL LINE SEGMENT OF A CIRCULAR TRIAL FAILURE SURFACE.	RANS 532 RANS 534 RANS 536 RANS 538 RANS 540


```

C      XINT      X-COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.      RANS 542
C      Y          Y COORDINATE OF THE MIDPOINT OF THE INITIAL LINE      RANS 544
C      Y          SEGMENT OF A TRIAL FAILURE SURFACE.                    RANS 558
C      YB         Y COORDINATE OF THE MIDPOINT OF A LINE SEGMENT WHOSE   RANS 560
C      YB         END POINTS ARE THE LEFT TERMINATION LIMIT AT THE       RANS 562
C      YB         GROUND SURFACE AND THE RIGHT END POINT OF THE INITIAL   RANS 564
C      YB         LINE SEGMENT OF A CIRCULAR TRIAL FAILURE SURFACE.      RANS 566
C      YBPT       Y COORDINATE OF LEFT TERMINATION LIMIT AT THE GROUND   RANS 568
C      YBPT       SURFACE.                                               RANS 570
C      YE         Y COORDINATE OF THE MIDPOINT OF A LINE SEGMENT WHOSE   RANS 572
C      YE         END POINTS ARE THE RIGHT TERMINATION LIMIT AT THE      RANS 574
C      YE         GROUND SURFACE AND THE RIGHT END POINT OF THE INITIAL   RANS 576
C      YE         LINE SEGMENT OF A CIRCULAR TRIAL FAILURE SURFACE.      RANS 578
C      YEPT       Y COORDINATE OF RIGHT TERMINATION LIMIT AT THE GROUND  RANS 580
C      YEPT       SURFACE.                                               RANS 582
C      YINT       Y-COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.    RANS 584
C      YMAX       END POINT OF SURFACE GENERATION BOUNDARY WITH THE     RANS 586
C      YMAX       MAXIMUM Y COORDINATE.                                  RANS 588
C      YMIN       LOWEST DEPTH TO WHICH A TRIAL SURFACE MAY EXTEND.      RANS 590
C      ----- RANS 592
C      ----- RANS 594
C      ----- RANS 596
C      ----- RANS 598
C      COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLOT,IREAD,ISEAPC,  RANS 600
C      1          IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL              RANS 602
C      COMMON /BLK02/BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND, RANS 604
C      1          NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20)              RANS 606
C      COMMON /BLK05/NSURF,SURF(100,2)                                  RANS 608
C      COMMON /BLK06/LIMIT(20,4),NLIMIT,NLMT                            RANS 610
C      COMMON /BLK12/ANGS1,ANGS2,BPT,EPT,FRTYFU,FS,FSS(12),JJ,NSURFS(12), RANS 612
C      1          PERPEN,SURFS(100,2,12),TSURF,YBPT,YEPT,YMIN          RANS 614
C      DIMENSION ERROR(4)                                              RANS 616
C      DATA ERROR/4HRC10,4HRC11,4HBK09,4HBK10/                      RANS 618
C      REAL LIMIT                                                       RANS 620
C      NR=0                                                             RANS 622
C      ----- RANS 624
C      ----- RANS 626
C      ----- RANS 628
C      ESTABLISH MINIMUM SLOPE OF FAILURE SURFACE                      RANS 630
C      ----- RANS 632
C      ----- RANS 634
C      ANGLMT=-FRTYFU                                                  RANS 636
C      IF (IANGL.EQ.1.AND.ANGS2.LT.ANGLMT)ANGLMT=ANGS2                RANS 638
C      3 IF (NR.GE.200)GO TO 13                                         RANS 640
C      NR=NR+1                                                         RANS 642
C      ----- RANS 644
C      ----- RANS 646
C      DETERMINE DIRECTION OF INITIAL SEGMENT AND CALCULATE           RANS 648
C      COORDINATES OF SECOND POINT FOR CIRCULAR OR IRREGULAR SURFACE  RANS 650
C      ----- RANS 652
C      ----- RANS 654
C      THETA=ANGS1-(ANGS1-ANGS2)*(1.-RANF(0.))**2)                    RANS 656
C      SURF(2,1)=SURF(1,1)+TSURF*COS(THETA)                            RANS 658
C      SURF(2,2)=SURF(1,2)+TSURF*SIN(THETA)                            RANS 660
C      ----- RANS 662
C      ----- RANS 664
C      ----- RANS 666
C      ----- RANS 668
C      ----- RANS 670
C      ----- RANS 672
C      ----- RANS 674
C      ----- RANS 676
C      ----- RANS 678
C      ----- RANS 680
C      ----- RANS 682
C      ----- RANS 684
C      ----- RANS 686
C      ----- RANS 688
C      ----- RANS 690

```


C	-----	RANS 692
C	CHECK FOR GROUND SURFACE INTERSECTION	RANS 694
C	-----	RANS 696
C		RANS 698
	DO 2 I=JJ,NTOP	RANS 700
	CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),SURF(1,1),	RANS 702
	1 SURF(1,2),SURF(2,1),SURF(2,2),XINT,YINT,INTS)	RANS 704
	IF(INTS.EQ.0)GO TO 2	RANS 706
	IF((BNDS(I,1)-SURF(1,1)).LE.TOL)GO TO 2	RANS 708
	SURF(2,1)=XINT	RANS 710
C		RANS 712
C	-----	RANS 714
C	IF INTERSECTION, CHECK FOR SATISFACTION OF REQUIREMENTS	RANS 716
C	-----	RANS 718
C		RANS 720
	IF(SURF(2,1).GT.BPT)GO TO 41	RANS 722
C		RANS 724
C	-----	RANS 726
C	IF INTERSECTION IS SHORT OF LEFT TERMINATION	RANS 728
C	LIMIT, CHANGE COUNTERCLOCKWISE DIRECTION	RANS 730
C	LIMIT FOR GENERATION OF FIRST LINE SEGMENT	RANS 732
C	-----	RANS 734
C		RANS 736
	ANGS1=THETA	RANS 738
	GO TO 3	RANS 740
	41 IF(SURF(2,1).LT.EPT)GO TO 42	RANS 742
C		RANS 744
C	-----	RANS 746
C	IF INTERSECTION BEYOND RIGHT TERMINATION	RANS 748
C	LIMIT, CHANGE CLOCKWISE DIRECTION LIMIT	RANS 750
C	FOR GENERATION OF FIRST LINE SEGMENT	RANS 752
C	-----	RANS 754
C		RANS 756
	ANGS2=THETA	RANS 758
	GO TO 3	RANS 760
	42 SURF(2,2)=YINT	RANS 762
	K=2	RANS 764
	IF(ILIMIT.EQ.0)GO TO 6	RANS 766
	DO 11 J=1,NLIMIT	RANS 768
	CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),	RANS 770
	1 SURF(1,1),SURF(1,2),SURF(2,1),SURF(2,2),XINT,YINT,INTS)	RANS 772
C		RANS 774
C	-----	RANS 776
C	IF LIMIT BOUNDARY INTERSECTED, DECREASE COUNTERCLOCKWISE	RANS 778
C	DIRECTION LIMIT IF BOUNDARY DEFLECTS DOWNWARD OR INCREASE	RANS 780
C	CLOCKWISE DIRECTION LIMIT IF BOUNDARY DEFLECTS UPWARD	RANS 782
C	-----	RANS 784
C		RANS 786
	IF(INTS.EQ.0)GO TO 11	RANS 788
	IF(I.LE.NLMT)ANGS2=THETA	RANS 790
	IF(I.GT.NLMT)ANGS1=THETA	RANS 792
	GO TO 3	RANS 794
	11 CONTINUE	RANS 796
	GO TO 6	RANS 798
	2 CONTINUE	RANS 800
C		RANS 802
C	-----	RANS 804
C	IF NO INTERSECTION, CHECK FOR SATISFACTION	RANS 806
C	OF INTERMEDIATE REQUIREMENTS	RANS 808
C	-----	RANS 810
C		RANS 812
	IF(SURF(2,2).GT.YMIN)GO TO 14	RANS 814
C		RANS 816

C	-----	RANS 818
C	IF INITIAL LINE SEGMENT EXTENDS BELOW DEPTH	RANS 820
C	LIMIT, CHANGE CLOCKWISE DIRECTION LIMIT	RANS 822
C	-----	RANS 824
C	ANGS2=THETA	RANS 826
	GO TO 3	RANS 828
	14 IF(ILIMIT.EQ.0)GO TO 8	RANS 830
	DO 7 I=1,NLIMIT	RANS 832
	CALL INTSCT(LIMIT(1,1),LIMIT(1,2),LIMIT(1,3),LIMIT(1,4),	RANS 834
	1 SURF(1,1),SURF(1,2),SURF(2,1),SURF(2,2),XINT,YINT,INTS)	RANS 836
	IF(INTS.EQ.0)GO TO 7	RANS 838
	-----	RANS 840
C	IF LIMIT BOUNDARY INTERSECTED, DECREASE COUNTERCLOCKWISE	RANS 842
C	DIRECTION LIMIT IF BOUNDARY DEFLECTS DOWNWARD OR INCREASE	RANS 844
C	CLOCKWISE DIRECTION LIMIT IF BOUNDARY DEFLECTS UPWARD	RANS 846
C	-----	RANS 848
	IF(I.LE.NLMT)ANGS2=THETA	RANS 850
	IF(I.GT.NLMT)ANGS1=THETA	RANS 852
	GO TO 3	RANS 854
	7 CONTINUE	RANS 856
	8 K=3	RANS 858
	-----	RANS 860
C	CHECK FOR CIRCULAR OR IRREGULAR SURFACE GENERATION	RANS 862
C	-----	RANS 864
	IF(ICIRC.EQ.0)GO TO 18	RANS 866
	-----	RANS 868
C	DETERMINE DEFLECTION LIMITS FOR	RANS 870
C	CORDS OF CIRCULAR SURFACE	RANS 872
C	-----	RANS 874
	THETAS=THETA	RANS 876
	THETAB=FRTYFU	RANS 878
	THETAE=TOL	RANS 880
	REY=1./TOL	RANS 882
	X=(SURF(1,1)+SURF(2,1))/2.	RANS 884
	Y=(SURF(1,2)+SURF(2,2))/2.	RANS 886
	TT=TAN(THETA)	RANS 888
	B=-1./TT	RANS 890
	A=Y-B*X	RANS 892
	SLPE=(YEPT-SURF(2,2))/(EPT-SURF(2,1))	RANS 894
	IF(SLPE.LE.TT)GO TO 23	RANS 896
	XE=(SURF(2,1)+EPT)/2.	RANS 898
	YE=(SURF(2,2)+YEPT)/2.	RANS 900
	BE=-1./SLPE	RANS 902
	AE=YE-BE*XE	RANS 904
	REX=(A-AE)/(BE-B)	RANS 906
	REY=A+B*REX	RANS 908
	RE=SQRT((REX-X)**2+(REY-Y)**2)	RANS 910
	THETAE=ATAN(TSURF/RE/2.)*2.	RANS 912
	23 IF(BPT.LE.SURF(1,1))GO TO 20	RANS 914
	SLPB=(YBPT-SURF(2,2))/(BPT-SURF(2,1))	RANS 916
	IF(SLPB.GT.TT)GO TO 53	RANS 918
	ANGS1=(YBPT-SURF(1,2))/(BPT-SURF(1,1))	RANS 920
	GO TO 3	RANS 922
	53 XB=(SURF(2,1)+BPT)/2.	RANS 924
	YB=(SURF(2,2)+YBPT)/2.	RANS 926
	BB=-1./SLPB	RANS 928
	AB=YB-BB*XB	RANS 930
	RBX=(A-AB)/(BB-B)	RANS 932
	RBX=A+B*RBX	RANS 934
	RB=SQRT((RBX-X)**2+(RBY-Y)**2)	RANS 936
	THETAB=ATAN(TSURF/RB/2.)*2.	RANS 938
		RANS 940
		RANS 942
		RANS 944
		RANS 946
		RANS 948
		RANS 950
		RANS 952

	IF(THETAB.GT.FRTYFU)THETAB=FRTYFU	RANS 954
	20 REY=REY-TSURF/2.-TOL	RANS 956
	NRR=0	RANS 957
	22 DTHETA=THETA+(THETAB-THETA)*RANF(0.)**2	RANS 958
	THETA=THETA	RANS 959
	NRR=NRR+1	RANS 960
	IF(NRR.GT.20)GO TO 3	RANS 961
C	-----	RANS 962
C	DETERMINE FOR CIRCULAR SURFACE THE DIRECTION OF	RANS 964
C	NEXT SEGMENT AND THE COORDINATES OF NEXT POINT	RANS 966
C	-----	RANS 968
	JT=JJ	RANS 970
	K=3	RANS 972
	17 THETA=THETA+DTHETA	RANS 974
C	-----	RANS 976
C	CHECK IF SURFACE EXCEEDS VERTICAL	RANS 978
C	-----	RANS 980
	IF((THETA+TOL).LT.PERPEN)GO TO 43	RANS 982
	IF(REY.GT.YEPT)GO TO 44	RANS 984
C	-----	RANS 986
C	INCREASE CLOCKWISE DIRECTION LIMIT OF INITIAL LINE SEGMENT	RANS 988
C	IF NO SURFACE SATISFIES VERTICAL SURFACE LIMITATION	RANS 990
C	-----	RANS 992
	ANGS2=THETA	RANS 994
	GO TO 3	RANS 996
	44 THETAB=DTHETA	RANS 998
	GO TO 22	RANS1000
	43 SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETA)	RANS1002
	IF(SURF(K,2).GT.YMIN)GO TO 47	RANS1004
	THETA=THETA	RANS1006
	I=3	RANS1008
	45 THETA=THETA+THETAB	RANS1010
	IF(THETA.GT.0.)GO TO 46	RANS1012
	SURF(I,2)=SURF(I-1,2)+TSURF*SIN(THETA)	RANS1014
	IF(SURF(I,2).LT.YMIN)GO TO 3	RANS1016
	I=I+1	RANS1018
	GO TO 45	RANS1020
	46 THETA=DTHETA	RANS1022
	GO TO 22	RANS1024
	47 SURF(K,1)=SURF(K-1,1)+TSURF*COS(THETA)	RANS1026
C	-----	RANS1028
C	CHECK FOR GROUND SURFACE INTERSECTION	RANS1030
C	-----	RANS1032
	DO 48 I=JT,NTOP	RANS1034
	ITN=I	RANS1036
	IF(SURF(K-1,1).GT.BNDS(I,3))GO TO 48	RANS1038
	CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),SURF(K-1,1),	RANS1040
	1 SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)	RANS1042
	IF(INTS.EQ.0)GO TO 55	RANS1044
	SURF(K,1)=XINT	RANS1046
	SURF(K,2)=YINT	RANS1048
C		RANS1050
		RANS1052
		RANS1054
		RANS1056
		RANS1058
		RANS1060
		RANS1062
		RANS1064
		RANS1066


```

C -----
C IF INTERSECTION, CHECK FOR SATISFACTION OF REQUIREMENTS
C -----
      IF(SURF(K,1).GE.BPT)GO TO 59
      THETA=THETAS
      II=2
      JTT=JT
63  II=II+1
      THETA=THETA+THETAЕ
      SURF(II,1)=SURF(II-1,1)+TSURF*COS(THETA)
      SURF(II,2)=SURF(II-1,2)+TSURF*SIN(THETA)
      DO 60 J=JTT,NTOP
      JTN=J
      IF(SURF(II-1,1).GT.BNDS(J,3))GO TO 60
      CALL INTSCT(BNDS(J,1),BNDS(J,2),BNDS(J,3),BNDS(J,4),SURF(II-1,1),
1     SURF(II-1,2),SURF(II,1),SURF(II,2),XINT,YINT,INTS)
      IF(INTS.EQ.0)GO TO 61
      IF(XINT.GT.BPT)GO TO 64
      ANGSI=THETAS
      GO TO 3
64  THETAB=DTHETA
      GO TO 22
61  IF(SURF(II,1).LT.BNDS(J+1,1))GO TO 62
60  CONTINUE
62  JTT=JTN
      GO TO 63
59  IF(ILIMIT.EQ.0)GO TO 6
      DO 49 J=1,NLIMIT
      CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),
1     SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)
      IF(INTS.EQ.1)GO TO 58
49  CONTINUE
      GO TO 6
55  IF(SURF(K,1).LT.BNDS(I+1,1))GO TO 54
48  CONTINUE
54  JT=ITN
C -----
C IF NO INTERSECTION, CHECK FOR SATISFACTION
C OF INTERMEDIATE REQUIREMENTS
C -----
      IF(ILIMIT.EQ.0)GO TO 50
      DO 51 J=1,NLIMIT
      CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),
1     SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)
      IF(INTS.EQ.1)GO TO 58
51  CONTINUE
50  K=K+1
C -----
C CHECK SURFACE POINT STORAGE LIMIT
C -----
      IF(K.LE.100)GO TO 17
      WRITE(6,102)ERROR(1)
102  FORMAT(/,10X,14H**** ERROR - ,A4,6H ****,/)
      CALL QUIT
58  THETA=THETAS
      YMAX=LIMIT(J,2)
      IF(LIMIT(J,4).GT.YMAX)YMAX=LIMIT(J,4)
      II=2
      IF(J.LE.NLMT)THET=THETAB-TOL
      IF(J.GT.NLMT)THET=THETAЕ+TOL

```

RANS1068
RANS1070
RANS1072
RANS1074
RANS1076
RANS1078
RANS1080
RANS1082
RANS1084
RANS1086
RANS1088
RANS1090
RANS1092
RANS1094 -
RANS1096
RANS1098
RANS1100
RANS1102
RANS1104
RANS1106
RANS1108
RANS1110
RANS1112
RANS1114
RANS1116
RANS1118 -
RANS1120
RANS1122
RANS1124
RANS1126
RANS1128
RANS1130
RANS1132
RANS1134
RANS1136
RANS1138
RANS1140 -
RANS1142
RANS1144
RANS1146
RANS1148
RANS1150
RANS1152
RANS1154
RANS1156
RANS1158
RANS1160
RANS1162
RANS1164
RANS1166
RANS1168
RANS1170
RANS1172
RANS1174
RANS1176
RANS1178
RANS1180
RANS1182
RANS1184
RANS1186
RANS1188
RANS1190
RANS1192
RANS1194
RANS1196


```

56 II=II+1                                RANS1198
   THETA=THETA+THET                       RANS1200
   SURF(II,1)=SURF(II-1,1)+TSURF*COS(THETA) RANS1202
   SURF(II,2)=SURF(II-1,2)+TSURF*SIN(THETA) RANS1204
   CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4), RANS1206
1   SURF(II-1,1),SURF(II-1,2),SURF(II,1),SURF(II,2),XINT,YINT,INTS) RANS1208
   IF(INTS.EQ.0)GO TO 57                   RANS1210
   IF(J.LE.NLMT)ANGS2=THETAS+TOL          RANS1212
   IF(J.GT.NLMT)ANGS1=THETAS-TOL          RANS1214
   GO TO 3                                 RANS1216
57 IF(SURF(II-1,2).GT.YMAX.AND.THETA.GT.0.)GO TO 65 RANS1218
   GO TO 56                               RANS1220
65 IF(J.LE.NLMT)THETA=DTTHETA              RANS1222
   IF(J.GT.NLMT)THETA=DTTHETA             RANS1224
   GO TO 22                               RANS1226
C                                         RANS1228
C ----- RANS1230
C DETERMINE DIRECTION LIMITS OF SUCCESSIVE RANS1232
C SEGMENTS FOR IRREGULAR SURFACE RANS1234
C ----- RANS1236
C RANS1238
18 THETA2=FRTYFU*RANF(0.):**2             RANS1240
   THETA1=THETA2+FRTYFU                   RANS1242
15 THETAU=THETA+FRTYFU                     RANS1244
   THETA1=THETAU-THETA1                   RANS1246
   IF((THETAU+TOL).GT.PERPEN)THETAU=PERPEN-TOL RANS1248
   IF(THETA1.LT.ANGLMT)THETA1=ANGLMT      RANS1250
C                                         RANS1252
C ----- RANS1254
C DETERMINE DIRECTION OF NEXT SEGMENT AND RANS1256
C CALCULATE COORDINATES OF NEXT POINT RANS1258
C ----- RANS1260
C RANS1262
4 THETA=THETA1+(THETAU-THETA1)*RANF(0.):**2 RANS1264
   SURF(K,1)=SURF(K-1,1)+TSURF*COS(THETA) RANS1266
   SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETA) RANS1268
C                                         RANS1270
C ----- RANS1272
C CHECK IF SURFACE EXTENDS BELOW MINIMUM RANS1274
C ----- RANS1276
C RANS1278
   IF(SURF(K,2).GT.YMIN)GO TO 16           RANS1280
   SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETAU) RANS1282
   IF(SURF(K,2).LT.YMIN)GO TO 3           RANS1284
   THETA1=THETA                           RANS1286
   GO TO 4                                 RANS1288
C                                         RANS1290
C ----- RANS1292
C CHECK FOR GROUND SURFACE INTERSECTION RANS1294
C ----- RANS1296
C RANS1298
16 DO 5 I=JJ,NTOP                          RANS1300
   CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),SURF(K-1,1), RANS1302
1   SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS) RANS1304
   IF(INTS.EQ.0)GO TO 5                   RANS1306
   SURF(K,1)=XINT                         RANS1308
C                                         RANS1310

```



```

C -----
C IF INTERSECTION, CHECK FOR SATISFACTION OF REQUIREMENTS
C -----
      IF(SURF(K,1).LT.EPT)GO TO 19
      SURF(K,1)=SURF(K-1,1)+TSURF*COS(THETAU)
      IF(SURF(K,1).GT.EPT)GO TO 3
      THETA=THETA
      GO TO 4
19 IF(SURF(K,1).LT.BPT)GO TO 3
      SURF(K,2)=YINT
      IF(ILIMIT.EQ.0)GO TO 6
      DO 12 J=1,NLIMIT
      CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),
1      SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)
      IF(INTS.EQ.0) GO TO 12
      IF(J.LE.NLMT) THETA=THETAU
      IF(J.GT.NLMT) THETA=THETA
      SURF(K,1)=SURF(K-1,1)+TSURF*COS(THETA)
      SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETA)
      IF(SURF(K,2).LT.YMIN)GO TO 3
      DO 21 L=1,NLIMIT
      CALL INTSCT(LIMIT(L,1),LIMIT(L,2),LIMIT(L,3),LIMIT(L,4),
1      SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)
      IF(INTS.EQ.1)GO TO 3
21 CONTINUE
      DO 68 N=JJ,NTOP
      CALL INTSCT(BNDS(N,1),BNDS(N,2),BNDS(N,3),BNDS(N,4),SURF(K-1,1),
1      SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)
      IF(INTS.EQ.0)GO TO 68
      SURF(K,1)=XINT
      SURF(K,2)=YINT
      IF(SURF(K,1).GT.EPT)GO TO 3
      GO TO 6
68 CONTINUE
      GO TO 9
12 CONTINUE
      GO TO 6
5 CONTINUE
      IF(SURF(K,1).GT.EPT)GO TO 3

C -----
C IF NO INTERSECTION, CHECK FOR SATISFACTION
C OF INTERMEDIATE REQUIREMENTS
C -----
      IF(ILIMIT.EQ.0)GO TO 9
      DO 10 I=1,NLIMIT
      CALL INTSCT(LIMIT(I,1),LIMIT(I,2),LIMIT(I,3),LIMIT(I,4),
1      SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)
      IF(INTS.EQ.0)GO TO 10
      IF(I.LE.NLMT) THETA=THETA
      IF(I.GT.NLMT) THETAU=THETA
      THETA=THETA+(THETAU-THETA)*RANF(0.)*(1.+RANF(0.))
      SURF(K,1)=SURF(K-1,1)+TSURF*COS(THETA)
      SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETA)
      DO 80 J=1,NLIMIT
      CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),
1      SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)
      IF(INTS.EQ.0) GO TO 80
      IF(J.LE.NLMT) THETA=THETAU
      IF(J.GT.NLMT) THETA=THETA
      SURF(K,1)=SURF(K-1,1)+TSURF*COS(THETA)
      SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETA)
      DO 81 L=1,NLIMIT
      CALL INTSCT(LIMIT(L,1),LIMIT(L,2),LIMIT(L,3),LIMIT(L,4),
1      SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)
      IF(INTS.EQ.1)GO TO 3
81 CONTINUE

```

RANS1312
RANS1314
RANS1316
RANS1318
RANS1320
RANS1322
RANS1324
RANS1326
RANS1328
RANS1330
RANS1332
RANS1334
RANS1336
RANS1338
RANS1340
RANS1342
RANS1344
RANS1346
RANS1348
RANS1350
RANS1352
RANS1354
RANS1356
RANS1358
RANS1360
RANS1362
RANS1364
RANS1366
RANS1368
RANS1370
RANS1372
RANS1374
RANS1376
RANS1378
RANS1380
RANS1382
RANS1384
RANS1386
RANS1388
RANS1390
RANS1392
RANS1394
RANS1396
RANS1398
RANS1400
RANS1402
RANS1404
RANS1406
RANS1408
RANS1410
RANS1412
RANS1414
RANS1416
RANS1418
RANS1420
RANS1422
RANS1424
RANS1426
RANS1428
RANS1430
RANS1432
RANS1434
RANS1436
RANS1438
RANS1440
RANS1442
RANS1444
RANS1446
RANS1448

GO TO 82	RANS1449
80 CONTINUE	RANS1450
82 DO 69 N=JJ,NTOP	RANS1452
CALL INTSCT(BNDS(N,1),BNDS(N,2),BNDS(N,3),BNDS(N,4),SURF(K-1,1),	RANS1454
1 SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)	RANS1456
IF(INTS.EQ.0)GO TO 69	RANS1458
SURF(K,1)=XINT	RANS1460
SURF(K,2)=YINT	RANS1462
IF(SURF(K,1).GT.EPT)GO TO 3	RANS1464
GO TO 6	RANS1466
69 CONTINUE	RANS1468
GO TO 9	RANS1469
10 CONTINUE	RANS1470
9 IF(SURF(K,1).GT.EPT)GO TO 3	RANS1472
IF(SURF(K,2).LT.YMIN)GO TO 3	RANS1474
K=K+1	RANS1476
-----	RANS1478
C CHECK SURFACE POINT STORAGE LIMIT	RANS1480
C	RANS1482
C	RANS1484
C	RANS1486
IF(K.LE.100)GO TO 15	RANS1488
WRITE(6,102)ERROR(1)	RANS1490
CALL QUIT	RANS1492
6 NSURF=K	RANS1494
RETURN	RANS1496
13 WRITE(6,102)ERROR(2)	RANS1498
IEXIT=2	RANS1502
RETURN	RANS1504
END	RANS1506

C	SUBROUTINE BLKSUF	BLKS	2
	-----	BLKS	4
C		BLKS	6
C	SUBROUTINE BLKSUF	BLKS	8
C	-----	BLKS	10
C		BLKS	12
C	FUNCTIONS -	BLKS	14
C		BLKS	16
C		BLKS	18
C	GENERATES TRIAL FAILURE SURFACES OF SLIDING BLOCK	BLKS	20
C	SHAPE BY A RANDOM TECHNIQUE.	BLKS	22
C		BLKS	24
C	-----	BLKS	26
C		BLKS	28
C	DEFINITIONS -	BLKS	30
C		BLKS	32
C	BNDS	AN ARRAY OF THE END COORDINATE POINTS DEFINING THE	BLKS 34
C		GROUND SURFACE AND SUBSURFACE BOUNDARIES.	BLKS 36
C			BLKS 38
C	COS	STANDARD FUNCTION THAT CALCULATES THE COSINE OF AN	BLKS 40
C		ANGLE.	BLKS 42
C			BLKS 44
C	ERROR	VARIABLE REPRESENTING CURRENT ERROR CODE.	BLKS 46
C			BLKS 48
C	FRTYFU	45 DEGREES IN RADIANS.	BLKS 50
C			BLKS 52
C	I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	BLKS 54
C			BLKS 56
C	ILIMIT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	BLKS 58
C		AFFECTED BY SEARCHING LIMITS ESTABLISHED BY	BLKS 60
C		SUBROUTINE LIMITS.	BLKS 62
C			BLKS 64
C	INTS	CONTROL CODE WHICH SIGNALS WHETHER AN INTERSECTION	BLKS 66
C		HAS OCCURRED OR NOT.	BLKS 68
C			BLKS 70
C	INTSCT	SUBROUTINE WHICH DETERMINES WHETHER OR NOT TWO LINE	BLKS 72
C		SEGMENTS INTERSECT AND CALCULATES THE COORDINATES OF	BLKS 74
C		THE INTERSECTION.	BLKS 76
C			BLKS 78
C	INTSC2	ENTRY POINT INTO SUBROUTINE INTSCT WHICH DETERMINES	NOU76EB0
C		WHETHER OR NOT A LINE SEGMENT AND A VERTICAL LINE	NOU76EB0
C		INTERSECT AND CALCULATES THE COORDINATES OF THE	NOU76EB0
C		INTERSECTION.	NOU76EB0
C			NOU76EB0
C	J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	BLKS 80
C			BLKS 82
C	JJ	DO LOOP COUNTER.	NOU76EB0
C			NOU76EB0
C	K	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	BLKS 84
C			BLKS 86
C	LIMIT	ARRAY CONTAINING THE COORDINATES OF THE END POINTS	BLKS 88
C		DEFINING EACH BOUNDARY USED TO CONFINE THE EXTENT OF	BLKS 90
C		SEARCHING BY BLOCK, RANDOM AND CIRCLE.	BLKS 92
C			BLKS 94
C	NGRID	NUMBER OF GRIDS SPECIFIED.	BLKS 96
C			BLKS 98
C	NLIMIT	NUMBER OF CONFINING BOUNDARIES SPECIFIED.	BLKS 100
C			BLKS 102
C	NR	TOTAL NUMBER OF REJECTIONS WHILE ATTEMPTING TO	BLKS 104
C		GENERATE A TRIAL FAILURE SURFACE.	BLKS 106
C			ELKS 108
C	NRR	NUMBER OF REJECTIONS WHILE ATTEMPTING TO GENERATE	BLKS 110
C		EITHER THE PASSIVE OR ACTIVE PORTIONS OF A SLIDING	BLKS 112
C		BLOCK TRIAL FAILURE SURFACE.	ELKS 114
C			BLKS 116
C	NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	BLKS 118
C			BLKS 120
C	NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	BLKS 122
C			BLKS 124

C	PLT4	ENTRY OF SUBROUTINE PLTN WHICH STORES THE PROFILE	BLKS 126
C		GEOMETRY AND, IF APPLICABLE, THE WATER SURFACE,	BLKS 128
C		SURFACE GENERATION BOUNDARIES, SURCHARGE BOUNDARY	BLKS 130
C		LOADS AND A SPECIFIED TRIAL FAILURE SURFACE. PLT4	BLKS 132
C		THEN PRINTS THE ENTIRE ARRAY AND ITS AXIS LABELS.	BLKS 134
C			BLKS 136
C	QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE,	BLKS 138
C		TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE	BLKS 140
C		PROGRAM.	BLKS 142
C			BLKS 144
C	R	VARIABLE CONTAINING A PSEUDO-RANDOM NUMBER.	BLKS 146
C			BLKS 148
C	RANF	FUNCTION SUBPROGRAM THAT GENERATES A PSEUDO-RANDOM	BLKS 150
C		NUMBER THAT HAS A UNIFORM PROBABILITY OF HAVING ANY	BLKS 152
C		VALUE RANGING FROM ZERO TO ONE.	BLKS 154
C			BLKS 156
C	SIN	STANDARD FUNCTION THAT CALCULATES THE SINE OF AN	BLKS 158
C		ANGLE.	BLKS 160
C			BLKS 162
C	SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS	BLKS 164
C		DEFINING A TRIAL FAILURE SURFACE.	BLKS 166
C			BLKS 168
C	SWITCH	ARRAY USED TO TEMPORARILY STORE THE CONTENTS OF ARRAY	BLKS 170
C		SURF FOR REORDERING.	BLKS 172
C			BLKS 174
C	THETA	DIRECTION OF LAST LINE SEGMENT DEFINING THE TRIAL	BLKS 176
C		SURFACE.	BLKS 178
C			BLKS 180
C	TSURF	LENGTH OF LINE SEGMENTS DEFINING TRIAL SURFACES.	BLKS 182
C			BLKS 184
C	WIDTH	ARRAY CONTAINING THE VALUES OF THE WIDTH OF EACH	BLKS 186
C		GRID.	BLKS 188
C			BLKS 190
C	XINT	X-COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.	BLKS 192
C			BLKS 194
C	XL	ARRAY CONTAINING VALUES OF THE X-COORDINATE OF THE	BLKS 196
C		LEFT END OF EACH GRID CENTERLINE.	BLKS 199
C			BLKS 200
C	XR	ARRAY CONTAINING VALUES OF THE X-COORDINATE OF THE	BLKS 202
C		RIGHT END OF EACH GRID CENTERLINE.	BLKS 204
C			BLKS 206
C	Y	Y COORDINATE OF VERTICAL INTERSECTION OF GROUND	BLKS 208
C		SURFACE FROM LAST POINT ON TRIAL FAILURE SURFACE.	BLKS 210
C			BLKS 212
C	YINT	Y-COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.	BLKS 214
C			BLKS 216
C	YL	ARRAY CONTAINING VALUES OF THE Y-COORDINATE OF THE	BLKS 218
C		LEFT END OF EACH GRID CENTERLINE.	BLKS 220
C			BLKS 222
C	YR	ARRAY CONTAINING VALUES OF THE Y-COORDINATE OF THE	BLKS 228
C		RIGHT END OF EACH GRID CENTERLINE.	BLKS 230
C			BLKS 232
C		-----	BLKS 234
C		-----	BLKS 236
C			BLKS 238
	COMMON /BLK01/	IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLT,IREAD,ISEARC.	BLKS 240
	1	IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL	BLKS 242
	COMMON /BLK02/	ENDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND,	BLKS 244
	1	NSOIL,NTOP,PHI(20),RU(20),CU(20),NF(20)	BLKS 246
	COMMON /BLK05/	NSURF,SURF(100,2)	BLKS 248
	COMMON /BLK06/	LIMIT(20,4),NLIMIT,NLMT	BLKS 250
	COMMON /BLK12/	ANGS1,ANGS2,EPT,EPT,FRTYFU,FS,FSS(12),JJ,NSURFS(12),	BLKS 252
	1	PERFEN,SURFS(100,2,12),TSURF,YBPT,YEPT,YMIN	BLKS 254
	COMMON /BLK13/	NGRID,WIDTH(10),XL(10),XR(10),YL(10),YR(10)	BLKS 256
		DIMENSION SWITCH(20,2),ERROR(4)	BLKS 258
		DATA ERROR/4HRC10,4HRC11,4HBK09,4HBK10/	BLKS 260
		REAL LIMIT	BLKS 262

	NR=0	BLKS 264
	26 J=NGRID	BLKS 266
C		BLKS 268
C	-----	BLKS 270
C	DEFINE A POINT WITHIN EACH BOX FOR THE TRIAL SURFACE	BLKS 272
C	-----	BLKS 274
C		BLKS 276
	DO 24 I=1,NGRID	BLKS 278
	R=RANF(0.)	BLKS 280
	SURF(J,1)=XL(I)+(XR(I)-XL(I))*R	BLKS 282
	SURF(J,2)=YL(I)+(YR(I)-YL(I))*R+WIDTH(I)*(RANF(0.)-.5)	BLKS 284
	J=J+1	BLKS 286
	24 CONTINUE	BLKS 288
C		BLKS 290
C	-----	BLKS 292
C	CHECK IF LAST POINT IS ON GROUND SURFACE	BLKS 294
C	-----	BLKS 296
C		BLKS 298
	K=NGRID	BLKS 300
	DO 66 I=1,NTOP	BLKS 302
	CALL INTSC2(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),0.,0.,0.,0.,	BLKS 304
	1SURF(K,1),Y,INTS)	BLKS 306
	IF(INTS.EQ.0) GO TO 66	BLKS 308
	IF (Y.EQ.SURF(K,2)) GO TO 29	BLKS 310
	66 CONTINUE	BLKS 312
	NRR=0	BLKS 314
	25 IF(NR.GE.200)GO TO 13	BLKS 316
	IF(NRR.GE.20)GO TO 26	BLKS 318
	NR=NR+1	BLKS 320
	NRR=NRR+1	BLKS 322
	K=NGRID+1	BLKS 324
C		BLKS 326
C	-----	BLKS 328
C	DETERMINE DIRECTION OF NEXT SEGMENT LEFT	BLKS 330
C	AND CALCULATE COORDINATES OF NEXT POINT	BLKS 332
C	-----	BLKS 334
C		BLKS 336
	27 THETA=(1.-RANF(0.))*2)*FRTYFU	BLKS 338
	SURF(K,1)=SURF(K-1,1)-TSURF*COS(THETA)	BLKS 340
	SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETA)	BLKS 342
C		BLKS 344
C	-----	BLKS 346
C	CHECK FOR GROUND SURFACE INTERSECTION	BLKS 348
C	-----	BLKS 350
C		BLKS 352
	I=NTOP+1	BLKS 354
	DO 28 JJ=1,NTOP	BLKS 356
	I=I-1	BLKS 358
	CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),SURF(K,1),	BLKS 360
	1 SURF(K,2),SURF(K-1,1),SURF(K-1,2),XINT,YINT,INTS)	BLKS 362
	IF(INTS.EQ.0)GO TO 28	BLKS 364
	SURF(K,1)=XINT	BLKS 366
	SURF(K,2)=YINT	BLKS 368
C		BLKS 370
C	-----	BLKS 372
C	IF INTERSECTION, CHECK FOR SATISFACTION OF REQUIREMENTS	BLKS 374
C	-----	BLKS 376
C		BLKS 378
	IF(SURF(K,1).LT.BNDS(1,1))GO TO 25	BLKS 380
	IF(ILIMIT.EQ.0)GO TO 29	BLKS 382
	DO 30 J=1,NLIMIT	BLKS 384
	CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),SURF(K,1),	BLKS 386
	1 SURF(K,2),SURF(K-1,1),SURF(K-1,2),XINT,YINT,INTS)	BLKS 388
	IF(INTS.NE.0)GO TO 25	BLKS 390
	30 CONTINUE	BLKS 392
	GO TO 29	BLKS 394
	28 CONTINUE	BLKS 396
C		BLKS 398

C	-----	BLKS 400
C	IF NO INTERSECTION, CHECK FOR SATISFACTION	BLKS 402
C	OF INTERMEDIATE REQUIREMENTS	BLKS 404
C	-----	BLKS 406
C	IF(SURF(K,1).LT.BNDS(1,1))GO TO 25	BLKS 408
	IF(ILIMIT.EQ.0)GO TO 31	BLKS 410
	DO 32 J=1,NLIMIT	BLKS 412
	CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),SURF(K,1),	BLKS 414
	1 SURF(K,2),SURF(K-1,1),SURF(K-1,2),XINT,YINT,INTS)	BLKS 416
	IF(INTS.NE.0)GO TO 25	BLKS 418
	32 CONTINUE	BLKS 420
	31 K=K+1	BLKS 422
C	-----	BLKS 424
C	CHECK SURFACE POINT STORAGE LIMIT	BLKS 426
C	-----	BLKS 428
C	IF(K.LE.100) GO TO 27	BLKS 430
	WRITE(6,102)ERROR(3)	BLKS 432
	102 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****, /)	BLKS 434
	CALL QUIT	BLKS 436
C	-----	BLKS 438
C	REVERSE ORDER OF POINTS GENERATED THUS FAR	BLKS 440
C	-----	BLKS 442
C	29 DO 33 I=1,K	BLKS 444
	SWITCH(I,1)=SURF(I,1)	BLKS 446
	SWITCH(I,2)=SURF(I,2)	BLKS 448
	33 CONTINUE	BLKS 450
	J=K	BLKS 452
	DO 34 I=1,K	BLKS 454
	SURF(I,1)=SWITCH(J,1)	BLKS 456
	SURF(I,2)=SWITCH(J,2)	BLKS 458
	J=J-1	BLKS 460
	34 CONTINUE	BLKS 462
	NSURF=K	BLKS 464
C	-----	BLKS 466
C	CHECK IF LAST POINT IS ON GROUND SURFACE	BLKS 468
C	-----	BLKS 470
C	DO 67 I=1,NTOP	BLKS 472
	CALL INTSC2(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),0.,0.,0.,0.,	BLKS 474
	1SURF(K,1),Y,INTS)	BLKS 476
	IF(INTS.EQ.0) GO TO 67	BLKS 478
	IF (Y.EQ.SURF(K,2)) GO TO 38	BLKS 480
	67 CONTINUE	BLKS 482
	NRR=0	BLKS 484
	35 IF(NR.GE.200)GO TO 13	BLKS 486
	IF(NRR.GE.20)GO TO 26	BLKS 488
	NR=NR+1	BLKS 490
	NRR=NRR+1	BLKS 492
	K=NSURF	BLKS 494
C	-----	BLKS 496
C	DETERMINE DIRECTION OF NEXT SEGMENT RIGHT	BLKS 498
C	AND CALCULATE COORDINATES OF NEXT POINT	BLKS 500
C	-----	BLKS 502
C	36 THETA=(1.-RANF(0.))**2)*FRYFU	BLKS 504
	K=K+1	BLKS 506
C		BLKS 508
		BLKS 510
		BLKS 512
		BLKS 514
		BLKS 516
		BLKS 518
		BLKS 520
		BLKS 522
		BLKS 524
		BLKS 526

C	-----	BLKS 528
C	CHECK SURFACE POINT STORAGE LIMIT	BLKS 530
C	-----	BLKS 532
C	IF(K.LE.100) GO TO 52	BLKS 534
	WRITE(6,102)ERROR(3)	BLKS 536
	CALL QUIT	BLKS 538
	52 SURF(K,1)=SURF(K-1,1)+TSURF*SIN(THETA)	BLKS 540
	SURF(K,2)=SURF(K-1,2)+TSURF*COS(THETA)	BLKS 542
C	-----	BLKS 544
C	CHECK FOR GROUND SURFACE INTERSECTION	BLKS 546
C	-----	BLKS 548
C	DO 37 I=1,NTOP	BLKS 550
	CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),SURF(K-1,1),	BLKS 552
	1 SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)	BLKS 554
	IF(INTS.EQ.0)GO TO 37	BLKS 556
	SURF(K,1)=XINT	BLKS 558
	SURF(K,2)=YINT	BLKS 560
C	-----	BLKS 562
C	IF INTERSECTION, CHECK FOR SATISFACTION OF REQUIREMENTS	BLKS 564
C	-----	BLKS 566
C	IF(SURF(K,1).GT.BNDS(NTOP,3))GO TO 35	BLKS 568
	IF(ILIMIT.EQ.0)GO TO 38	BLKS 570
	DO 39 J=1,NLIMIT	BLKS 572
	CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),	BLKS 574
	1 SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)	BLKS 576
	IF(INTS.NE.0)GO TO 35	BLKS 578
	39 CONTINUE	BLKS 580
	GO TO 38	BLKS 582
	37 CONTINUE	BLKS 584
C	-----	BLKS 586
C	IF NO INTERSECTION, CHECK FOR SATISFACTION	BLKS 588
C	OF INTERMEDIATE REQUIREMENTS	BLKS 590
C	-----	BLKS 592
C	IF(SURF(K,1).GT.BNDS(NTOP,3))GO TO 35	BLKS 594
	IF(ILIMIT.EQ.0)GO TO 36	BLKS 596
	DO 40 J=1,NLIMIT	BLKS 598
	CALL INTSCT(LIMIT(J,1),LIMIT(J,2),LIMIT(J,3),LIMIT(J,4),	BLKS 600
	1 SURF(K-1,1),SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)	BLKS 602
	IF(INTS.NE.0)GO TO 35	BLKS 604
	40 CONTINUE	BLKS 606
	GO TO 36	BLKS 608
	38 NSURF=K	BLKS 610
	RETURN	BLKS 612
	13 WRITE(6,102)ERROR(4)	BLKS 614
	IEXIT=2	BLKS 616
	RETURN	BLKS 618
	END	BLKS 620
		BLKS 622
		BLKS 624
		BLKS 626
		BLKS 628
		BLKS 632
		BLKS 634
		BLKS 636

C	SUBROUTINE BLOCK2	BK2-	2
C	-----	BK2-	4
C	-----	BK2-	6
C	SUBROUTINE BLOCK2	BK2-	8
C	-----	BK2-	10
C	-----	BK2-	12
C	FUNCTIONS -	BK2-	14
C		BK2-	16
C		BK2-	18
C	GENERATES TRIAL FAILURE SURFACES OF SLIDING BLOCK SHAPE	BK2-	20
C	BY A RANDOM TECHNIQUE. ACTIVE AND PASSIVE PORTIONS ARE	BK2-	22
C	GENERATED ACCORDING TO THE RANKINE THEORY.	BK2-	24
C	-----	BK2-	26
C	-----	BK2-	28
C	DEFINITIONS -	BK2-	30
C		BK2-	32
C		BK2-	34
C	BNDS AN ARRAY OF THE END COORDINATE POINTS DEFINING THE	BK2-	36
C	GROUND SURFACE AND SUBSURFACE BOUNDARIES.	BK2-	38
C		BK2-	40
C	COS STANDARD FUNCTION THAT CALCULATES THE COSINE OF AN	BK2-	42
C	ANGLE.	BK2-	44
C		BK2-	46
C	ERROR VARIABLE REPRESENTING CURRENT ERROR CODE.	BK2-	48
C		BK2-	50
C	FRTYFU 45 DEGREES IN RADIANS.	BK2-	52
C		BK2-	54
C	I INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	BK2-	56
C		BK2-	58
C	INTS CONTROL CODE WHICH SIGNALS WHETHER AN INTERSECTION	BK2-	60
C	HAS OCCURRED OR NOT.	BK2-	62
C		BK2-	64
C	INTSCT SUBROUTINE WHICH DETERMINES WHETHER OR NOT TWO LINE	BK2-	66
C	SEGMENTS INTERSECT AND CALCULATES THE COORDINATES OF	BK2-	68
C	THE INTERSECTION.	BK2-	70
C		BK2-	72
C	INTSC2 ENTRY POINT INTO SUBROUTINE INTSCT WHICH DETERMINES	BK2-	74
C	WHETHER OR NOT A LINE SEGMENT AND A VERTICAL LINE	BK2-	76
C	INTERSECT AND CALCULATES THE COORDINATES OF THE	BK2-	78
C	INTERSECTION.	BK2-	80
C		BK2-	82
C	ITP ARRAY CONTAINING SOIL TYPE INDICES FOR EACH BOUNDARY.	BK2-	84
C		BK2-	86
C	J INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	BK2-	88
C		BK2-	90
C	JAB SUBSCRIPT OF THE LAST PROFILE BOUNDARY USED TO CHECK	BK2-	100
C	FOR INTERSECTION WITH THE LAST LINESEGMENT GENERATED	BK2-	102
C	ON THE ACTIVE PORTION OF THE TRIAL FAILURE SURFACE.	BK2-	104
C		BK2-	106
C	JJ INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	BK2-	108
C		BK2-	110
C	JP SUBSCRIPT OF THE LAST PROFILE BOUNDARY USED TO CHECK	BK2-	112
C	FOR INTERSECTION WITH VERTICAL LINE THROUGH THE LAST	BK2-	114
C	POINT GENERATED ON THE TRIAL FAILURE SURFACE.	BK2-	116
C		BK2-	118
C	JPB SUBSCRIPT OF THE LAST PROFILE BOUNDARY USED TO CHECK	BK2-	120
C	FOR INTERSECTION WITH THE LAST LINESEGMENT GENERATED	BK2-	122
C	ON THE PASSIVE PORTION OF THE TRIAL FAILURE SURFACE.	BK2-	124
C		BK2-	126
C	K INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	BK2-	128
C		BK2-	130
C	M CONTROL NUMBER FOR ERROR CODE IN SUBROUTINE BLOCK2	AUG76EB0	
C		AUG76EB0	
C	NBND TOTAL NUMBER OF PROFILE BOUNDARIES.	BK2-	132
C		BK2-	134
C	NGRID NUMBER OF GRIDS SPECIFIED.	BK2-	136
C		BK2-	138

C C C C C C C C C C C C C C C C C C C C C C C C C C C C C	NR	TOTAL NUMBER OF REJECTIONS WHILE ATTEMPTING TO GENERATE A TRIAL FAILURE SURFACE.	BK2- 140 BK2- 142 BK2- 144
	NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	BK2- 146 BK2- 148
	NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	BK2- 150 BK2- 152
	PHI	ARRAY CONTAINING ISOTROPIC VALUES OF THE MOHR-COULOMB ANGLE OF INTERNAL FRICTION FOR EACH SOIL TYPE.	BK2- 154 BK2- 156 BK2- 158
	PLT4	ENTRY OF SUBROUTINE PLTN WHICH STORES THE PROFILE GEOMETRY AND, IF APPLICABLE, THE WATER SURFACE, SURFACE GENERATION BOUNDARIES, SURCHARGE BOUNDARY LOADS AND A SPECIFIED TRIAL FAILURE SURFACE. PLT4 THEN PRINTS THE ENTIRE ARRAY AND ITS AXIS LABLES.	BK2- 160 BK2- 162 BK2- 164 BK2- 166 BK2- 168 BK2- 170
	QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE, TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE PROGRAM.	BK2- 172 BK2- 174 BK2- 176 BK2- 178
	R	VARIABLE CONTAINING A PSEUDO-RANDOM NUMBER.	BK2- 180 BK2- 182
	RANF	FUNCTION SUBPROGRAM THAT GENERATES A PSEUDO-RANDOM NUMBER THAT HAS A UNIFORM PROBABILITY OF HAVING ANY VALUE RANGING FROM ZERO TO ONE.	BK2- 184 BK2- 186 BK2- 188 BK2- 190
	RD	FACTOR FOR CONVERSION OF DEGREES TO RADIANS.	BK2- 192 BK2- 194
	SIN	STANDARD FUNCTION THAT CALCULATES THE SINE OF AN ANGLE.	BK2- 196 BK2- 198 BK2- 200
	SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS DEFINING A TRIAL FAILURE SURFACE.	BK2- 202 BK2- 204 BK2- 206
	SWITCH	ARRAY USED TO TEMPORARILY STORE THE CONTENTS OF ARRAY SURF FOR REORDERING.	BK2- 208 BK2- 210 BK2- 212
	THETA	DIRECTION OF LAST LINE SEGMENT DEFINING THE TRIAL SURFACE.	BK2- 214 BK2- 216 BK2- 218
	TSURF	LENGTH OF LINE SEGMENTS DEFINING TRIAL SURFACES.	BK2- 220 BK2- 222
	WIDTH	ARRAY CONTAINING THE VALUES OF THE WIDTH OF EACH GRID.	BK2- 224 BK2- 226 BK2- 228
	XINT	X-COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.	BK2- 230 BK2- 232
	XL	ARRAY CONTAINING VALUES OF THE X-COORDINATE OF THE LEFT END OF EACH GRID CENTERLINE.	BK2- 234 BK2- 236 BK2- 238
	XR	ARRAY CONTAINING VALUES OF THE X-COORDINATE OF THE RIGHT END OF EACH GRID CENTERLINE.	BK2- 240 BK2- 242 BK2- 244
	Y	Y COORDINATE OF VERTICAL INTERSECTION OF GROUND SURFACE FROM LAST POINT ON TRIAL FAILURE SURFACE.	BK2- 246 BK2- 248 BK2- 250
	YINT	Y-COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.	BK2- 252 BK2- 254
	YL	ARRAY CONTAINING VALUES OF THE Y-COORDINATE OF THE LEFT END OF EACH GRID CENTERLINE.	BK2- 256 BK2- 258 BK2- 260
	YR	ARRAY CONTAINING VALUES OF THE Y-COORDINATE OF THE RIGHT END OF EACH GRID CENTERLINE.	BK2- 266 BK2- 268 BK2- 270 BK2- 272 BK2- 274


```

C
COMMON /BLK01/ IANGL, IBLK, IEXIT, ICIRC, ILIMIT, IPLOT, IREAD, ISEARC,
1 IBLK2, ISOIL, ISTR, ISUPC, ISURF, IWAT, RD, TOL
COMMON /BLK02/ BNDS(100,4), C(20), GAMMA(20), GSAT(20), ITP(100), NE'D,
1 NSOIL, NTOP, PHI(20), RU(20), CU(20), NP(20)
COMMON /BLK05/ NSURF, SURF(100,2)
COMMON /BLK06/ LIMIT(20,4), NLIMIT, NLMT
COMMON /BLK12/ ANGS1, ANGS2, BPT, EPT, FRTYFU, FS, FSS(12), JJ, NSURFS(12),
1 PERPEN, SURFS(100,2,12), TSURF, YBPT, YEPT, YMIN
COMMON /BLK13/ NGRID, WIDTH(10), XL(10), XR(10), YL(10), YR(10)
COMMON /BLK15/ M, MB
DIMENSION SWITCH(20,2), ERROR(5)
DATA ERROR/4HRC10, 4HRC11, 4HBK09, 4HBK10, 4HBK12/
NR=0
26 J=NGRID
NR=NR+1
IF(NR.GE.200)GO TO 13

C
C
C -----
C DEFINE A POINT WITHIN EACH BOX FOR THE TRIAL SURFACE
C -----
C
DO 24 I=1,NGRID
R=РАНF(0.)
SURF(J,1)=XL(I)+(XR(I)-XL(I))*R
SURF(J,2)=YL(I)+(YR(I)-YL(I))*R+WIDTH(I)*(РАНF(0.)-.5)
J=J+1
24 CONTINUE

C
C -----
C CHECK IF LAST POINT IS ON GROUND SURFACE
C -----
C
K=NGRID
DO 66 I=1,NTOP
CALL INTSC2(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),0.,0.,0.,0.,
1SURF(K,1),Y,INTS)
IF(INTS.EQ.0) GO TO 66
IF (Y.EQ.SURF(K,2)) GO TO 29
66 CONTINUE

C
C -----
C PASSIVE WEDGE
C -----
C
K=NGRID
JP=0
JPB=0
70 I=NBND-JP+1
IF(I.LE.1) GO TO 29
J=NBND-JP
DO 71 JJ=1,J
I=I-1
CALL INTSC2(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),0.,0.,0.,0.,
1SURF(K,1),Y,INTS)
IF(INTS.EQ.0) GO TO 71
IF(Y.LE.SURF(K,2)) GO TO 71
THETA=FRTYFU-PHI(ITP(I))/2.*RD
JP=JP+JJ
GO TO 72
71 CONTINUE
GO TO 29
72 K=K+1
SURF(K,1)=SURF(K-1,1)-TSURF*COS(THETA)
SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETA)
I=NBND-JPB
J=I

```

BK2- 276
BK2- 278
BK2- 280
BK2- 282
BK2- 284
BK2- 286
BK2- 288
BK2- 290
BK2- 292
BK2- 294
BK2- 295
BK2- 296
BK2- 298
BK2- 300
BK2- 302
BK2- 304
BK2- 306
BK2- 308
BK2- 310
BK2- 312
BK2- 314
BK2- 316
BK2- 318
BK2- 320
BK2- 322
BK2- 324
BK2- 326
BK2- 328
BK2- 330
BK2- 332
BK2- 334
BK2- 336
BK2- 338
BK2- 340
BK2- 342
BK2- 344
BK2- 346
BK2- 348
BK2- 350
BK2- 352
BK2- 354
BK2- 356
BK2- 358
BK2- 360
BK2- 362
BK2- 364
BK2- 366
BK2- 368
BK2- 370
BK2- 372
BK2- 373
BK2- 374
BK2- 376
BK2- 378
BK2- 380
BK2- 382
BK2- 384
BK2- 386
BK2- 388
BK2- 390
BK2- 392
BK2- 394
BK2- 396
BK2- 398
BK2- 400
BK2- 402
BK2- 403

DO 73 JJ=1,J	BK2- 404
CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),SURF(K,1),	BK2- 406
1 SURF(K,2),SURF(K-1,1),SURF(K-1,2),XINT,YINT,INTS)	BK2- 408
I=I-1	BK2- 410
IF(INTS.EQ.0) GO TO 73	BK2- 412
SURF(K,1)=XINT	BK2- 414
SURF(K,2)=YINT	BK2- 416
JPB=JPB+JJ	BK2- 418
-----	BK2- 420
CHECK SURFACE POINT STORAGE LIMIT	BK2- 422
-----	BK2- 424
IF(K.LE.99) GO TO 70	BK2- 428
WRITE(6,102)ERROR(3)	BK2- 430
102 FORMAT(/,10X,14H**** ERROR - ,A4,6H ****, /)	BK2- 432
CALL QUIT	BK2- 434
73 CONTINUE	BK2- 436
IF(SURF(K,1).GE.BNDS(1,1)) GO TO 72	BK2- 438
M=M+1	BK2- 440
IF(M.EQ.1) WRITE(6,102)ERROR(5)	BK2- 441
GO TO 26	BK2- 442
-----	BK2- 444
REVERSE ORDER OF POINTS GENERATED THUS FAR	BK2- 446
-----	BK2- 448
29 DO 33 I=1,K	BK2- 450
SWITCH(I,1)=SURF(I,1)	BK2- 452
SWITCH(I,2)=SURF(I,2)	BK2- 454
33 CONTINUE	BK2- 456
J=K	BK2- 458
DO 34 I=1,K	BK2- 460
SURF(I,1)=SWITCH(J,1)	BK2- 462
SURF(I,2)=SWITCH(J,2)	BK2- 464
J=J-1	BK2- 466
34 CONTINUE	BK2- 468
-----	BK2- 470
CHECK IF LAST POINT IS ON GROUND SURFACE	BK2- 472
-----	BK2- 474
DO 67 I=1,NTOP	BK2- 476
CALL INTSC2(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),0.,0.,0.,0.,	BK2- 478
1SURF(K,1),Y,INTS)	BK2- 480
IF(INTS.EQ.0) GO TO 67	BK2- 482
IF (Y.EQ.SURF(K,2)) GO TO 38	BK2- 484
67 CONTINUE	BK2- 486
-----	BK2- 488
ACTIVE WEDGE	BK2- 490
-----	BK2- 492
JAB=0	BK2- 494
74 I=NBND+1	BK2- 496
IF(I.LE.1) GO TO 38	BK2- 498
DO 75 JJ=1,NBND	BK2- 500
I=I-1	BK2- 502
CALL INTSC2(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),0.,0.,0.,0.,	BK2- 504
1SURF(K,1),Y,INTS)	BK2- 506
IF(INTS.EQ.0) GO TO 75	BK2- 510
IF(Y.LE.SURF(K,2)) GO TO 75	BK2- 512
THETA=FRYFU+PHI(ITP(I))/2.*RD	BK2- 514
GO TO 76	BK2- 516
75 CONTINUE	BK2- 518
GO TO 38	BK2- 520

76	K=K+1	BK2-	538
	SURF(K,1)=SURF(K-1,1)+TSURF*COS(THETA)	BK2-	540
	SURF(K,2)=SURF(K-1,2)+TSURF*SIN(THETA)	BK2-	542
	I=NBND-JAB	BK2-	544
	J=I	BK2-	545
	DO 77 JJ=1,J	BK2-	546
	CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),SURF(K-1,1),	BK2-	548
1	SURF(K-1,2),SURF(K,1),SURF(K,2),XINT,YINT,INTS)	BK2-	550
	I=I-1	BK2-	552
	IF(INTS.EQ.0) GO TO 77	BK2-	554
	SURF(K,1)=XINT	BK2-	556
	SURF(K,2)=YINT	BK2-	558
	JAB=JAB+JJ	BK2-	560
		BK2-	562
	-----	BK2-	564
	CHECK SURFACE POINT STORAGE LIMIT	BK2-	566
	-----	BK2-	568
		BK2-	570
	IF(K.LE.99) GO TO 74	BK2-	572
	WRITE(6,102)ERROR(3)	BK2-	574
	CALL QUIT	BK2-	576
77	CONTINUE	BK2-	578
	IF(SURF(K,1).LE.BNDS(NTOP,3)) GO TO 76	BK2-	580
	M=M+1	BK2-	581
	IF(M.EQ.1) WRITE(6,102)ERROR(5)	BK2-	582
	GO TO 26	BK2-	584
38	NSURF=K	BK2-	586
	RETURN	BK2-	588
13	WRITE(6,102)ERROR(4)	BK2-	590
	IEXIT=2	BK2-	594
	RETURN	BK2-	596
	END	BK2-	598

	SUBROUTINE SORT	SORT 2
C	-----	SORT 4
C		SORT 6
C	SUBROUTINE SORT	SORT 8
C	-----	SORT 10
C		SORT 12
C		SORT 14
C	FUNCTIONS -	SORT 16
C		SORT 18
C	SORTS ARRAYS CONTAINING THE COORDINATES OF THE TEN MOST	SORT 20
C	CRITICAL FAILURE SURFACES ACCUMULATED THUS FAR AND THEIR	SORT 22
C	FACTORS OF SAFETY IN ORDER OF THE MAGNITUDE OF THEIR FACTORS	SORT 24
C	OF SAFETY.	SORT 26
C		SORT 28
C	-----	SORT 30
C		SORT 32
C	DEFINITIONS -	SORT 34
C		SORT 36
C	FSS ARRAY CONTAINING VALUES OF THE FACTOR OF SAFETY FOR	SORT 38
C	THE TEN MOST CRITICAL TRIAL SURFACES OF THOSE	SORT 40
C	GENERATED AND EXAMINED.	SORT 42
C		SORT 44
C	J INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	SORT 46
C		SORT 48
C	K INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	SORT 50
C		SORT 52
C	L COUNTER FOR DO LOOP.	SORT 54
C		SORT 56
C	N VARIABLE USED TO TEMPORARILY STORE THE NUMBER OF	SORT 58
C	POINTS DEFINING A TRIAL FAILURE SURFACE.	SORT 60
C		SORT 62
C	NSURFS ARRAY CONTAINING VALUES OF THE NUMBER OF POINTS	SORT 64
C	DEFINING EACH OF THE TEN MOST CRITICAL TRIAL	SORT 66
C	SURFACES.	SORT 68
C		SORT 70
C	SURFS ARRAY CONTAINING SETS OF COORDINATE POINTS DEFINING	SORT 72
C	EACH OF THE TEN MOST CRITICAL TRIAL SURFACES.	SORT 74
C		SORT 76
C	-----	SORT 78
C		SORT 80
C	COMMON /BLK12/ANGS1,ANGS2,BPT,EPT,FRTYFU,FS,FSS(12),JJ,NSURFS(12),	SORT 82
C	1 PERPEN,SURFS(100,2,12),TSURF,YBPT,YEPT,YMIN	SORT 84
	DO 15 L=1,10	SORT 86
	DO 15 K=2,10	SORT 88
	IF(FSS(K-1).LE.FSS(K))GO TO 15	SORT 90
	N=NSURFS(K)	SORT 92
	DO 16 J=1,N	SORT 94
	SURFS(J,1,12)=SURFS(J,1,K)	SORT 96
	SURFS(J,2,12)=SURFS(J,2,K)	SORT 98
16	CONTINUE	SORT 100
	NSURFS(12)=NSURFS(K)	SORT 102
	FSS(12)=FSS(K)	SORT 104
	N=NSURFS(K-1)	SORT 106
	DO 19 J=1,N	SORT 108
	SURFS(J,1,K)=SURFS(J,1,K-1)	SORT 110
	SURFS(J,2,K)=SURFS(J,2,K-1)	SORT 112
19	CONTINUE	SORT 114
	FSS(K)=FSS(K-1)	SORT 116
	NSURFS(K)=NSURFS(K-1)	SORT 118
	N=NSURFS(12)	SORT 120
	DO 17 J=1,N	SORT 122
	SURFS(J,1,K-1)=SURFS(J,1,12)	SORT 124
	SURFS(J,2,K-1)=SURFS(J,2,12)	SORT 126
17	CONTINUE	SORT 128
	FSS(K-1)=FSS(12)	SORT 130
	NSURFS(K-1)=NSURFS(12)	SORT 132
15	CONTINUE	SORT 134
	RETURN	SORT 136
	END	SORT 138
		SORT 140


```

C SUBROUTINE EXECUT EXEC 2
C ----- EXEC 4
C ----- EXEC 6
C SUBROUTINE EXECUT EXEC 8
C ----- EXEC 10
C ----- EXEC 12
C EXEC 14
C FUNCTIONS - EXEC 16
C EXEC 18
C CONTROLS CALCULATION OF THE FACTOR OF SAFETY AND INTERMEDIATE EXEC 20
C STEPS. EXEC 22
C EXEC 24
C IF SINGLE TRIAL FAILURE SURFACE SPECIFIED, IMPLEMENTS PLOT EXEC 26
C OF SURFACE AND PROBLEM GEOMETRY. EXEC 28
C EXEC 30
C ----- EXEC 32
C EXEC 34
C DEFINITIONS - EXEC 36
C EXEC 38
C FACTR SUBROUTINE WHICH CALCULATES THE FACTOR OF SAFETY. EXEC 40 -
C EXEC 42
C ISEARC CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM EXEC 50
C AFFECTED BY THE SEARCHING SUBROUTINES RANDOM AND
C BLOCK. EXEC 52
C EXEC 54
C EXEC 56
C PLOTIN SUBROUTINE WHICH PLOTS WITH A PLOTTING DEVICE THE EXEC 58
C PROFILE, AND IF SPECIFIED, THE WATER SURFACE, EXEC 60
C SEARCHING BOUNDARIES, AND A SPECIFIED TRIAL FAILURE EXEC 62
C SURFACE. EXEC 64
C EXEC 66
C PLT4 SUBROUTINE ENTRY OF SUBROUTINE PLTN WHICH PLOTS WITH EXEC 74
C PRINT CHARACTERS THE PROFILE BOUNDARY END POINTS, AND EXEC 76
C IF SPECIFIED, WATER SURFACE POINTS, SEARCHING LIMIT EXEC 78
C END POINTS, AND SPECIFIED TRIAL FAILURE SURFACE EXEC 80
C POINTS. EXEC 82
C EXEC 84
C SLICES SUBROUTINE WHICH DIVIDES SLIDING MASS INTO SLICES. EXEC 98
C EXEC 100
C WEIGHT SUBROUTINE WHICH CALCULATES THE TOTAL WEIGHT OF EACH EXEC 102
C SLICE. EXEC 104
C EXEC 106
C ----- EXEC 108
C ----- EXEC 110
C EXEC 112
C COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLT,IREAD,ISEARC, EXEC 114
C 1 IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL EXEC 116 -
C EXEC 118
C ----- EXEC 120
C ACTIVATE PORTIONS OF THE PROGRAM THAT CALCULATES THE FACTOR EXEC 122
C OF SAFETY FOR A SPECIFIED OR GENERATED TRIAL FAILURE SURFACE EXEC 124
C ----- EXEC 126
C EXEC 128
C CALL SLICES EXEC 130
C CALL WEIGHT EXEC 132
C CALL FACTR EXEC 134
C IF(ISEARC.EQ.1)RETURN EXEC 136
C EXEC 138
C ----- EXEC 140
C IF SINGLE TRIAL FAILURE SURFACE SPECIFIED, IMPLEMENT EXEC 142
C PLOT OF THE SURFACE AND PROFILE GEOMETRY EXEC 144
C ----- EXEC 146
C EXEC 148
C CALL PLT4 EXEC 150
C CALL PLOTIN EXEC 152
C RETURN EXEC 154
C END EXEC 156

```


C	N	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	JUL76E80 -
C	NBND	TOTAL NUMBER OF PROFILE BOUNDARIES.	JUL76E80 -
C	NN	VARIABLE USED TO TEMPORARY STORE AN ARRAY VARIABLE FOR A DO STATEMENT.	SLIC 128
C			SLIC 130
C	NPIEZ	NUMBER OF POINTS DEFINING THE WATER SURFACE.	AUG76E80
C			AUG76E80
C			AUG76E80
C	NPZ	NUMBER OF PIEZOMETRIC SURFACES DEFINED.	SLIC 132
C			SLIC 134
C	NSLICE	NUMBER OF SLICES WHICH THE SLIDING MASS IS DIVIDED.	JUL76E80 -
C			JUL76E80 -
C	NSURC	NUMBER OF BOUNDARY LOADS SPECIFIED.	SLIC 136
C			SLIC 138
C	NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	SLIC 140
C			SLIC 142
C	NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	SLIC 144
C			SLIC 146
C	NTOP1	NTOP + 1	SLIC 148
C			SLIC 150
C			SLIC 152
C	SURC	ARRAY CONTAINING X COORDINATES OF THE END POINTS DEFINING THE EXTENT OF LOADING.	SLIC 154
C			SLIC 156
C			SLIC 158
C	SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS DEFINING A TRIAL FAILURE SURFACE.	SLIC 160
C			SLIC 162
C			SLIC 164
C			SLIC 166
C	TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	SLIC 168
C			SLIC 170
C	X	ARRAY CONTAINING POSITIONS OF THE DIVISIONS BETWEEN SLICES.	SLIC 172
C			SLIC 174
C	XINT	X COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.	SLIC 176
C			SLIC 178
C	XPIEZ	ARRAY CONTAINING X COORDINATES OF POINTS DEFINING WATER SURFACE.	SLIC 180
C			SLIC 182
C			SLIC 184
C			SLIC 186
C	Y	Y COORDINATE OF THE TRIAL FAILURE SURFACE IMMEDIATELY ABOVE OR BELOW AN END POINT OF A SUBSURFACE BOUNDARY.	SLIC 188
C			SLIC 190
C			SLIC 192
C	YINT	Y COORDINATE OF INTERSECTION OF TWO LINE SEGMENTS.	SLIC 194
C			SLIC 196
C	YPIEZ	ARRAY CONTAINING Y COORDINATES OF POINTS DEFINING WATER SURFACE.	SLIC 198
C			SLIC 200
C			SLIC 202
C			SLIC 204
C			SLIC 206
C			SLIC 208
C	COMMON /BLK01/	IANGI, IBLK, IEXIT, ICIRC, ILIMIT, IPLOT, IREAD, ISEARC,	SLIC 210
C	1	IBLK2, ISOIL, ISTR, ISURC, ISURF, IWAT, RD, TOL	SLIC 212 -
C	COMMON /BLK02/	BNDS(100,4), C(20), GAMMA(20), GSAT(20), ITP(100), NBND,	SLIC 214
C	1	NSOIL, NTOP, PHI(20), RU(20), CU(20), NP(20)	SLIC 216 -
C	COMMON /BLK03/	UHAT, NPZ, NPIEZ(10), XPIEZ(10,40), YPIEZ(10,40)	SLIC 218 -
C	COMMON /BLK04/	DELTA(10), LOAD(10), NSURC, SURC(10,2), YSURC(10,2)	SLIC 220
C	COMMON /BLK05/	NSURF, SURF(100,2)	SLIC 222
C	COMMON /BLK08/	NSLICE, X(300)	SLIC 224
C		K=0	SLIC 226
C			SLIC 228
C			SLIC 230
C			SLIC 232
C			SLIC 234
C			SLIC 236
C			SLIC 238
C			SLIC 240
C			SLIC 242
C			SLIC 244
C			SLIC 246
C			SLIC 248

DIVIDE SLIDING MASS AT EACH POINT
DEFINING THE TRIAL FAILURE SURFACE

DO 3 I=1, NSURF
K=K+1
X(K)=SURF(I,1)
3 CONTINUE

SLIC 232

SLIC 234

SLIC 236

SLIC 238

SLIC 240

SLIC 242

SLIC 244

SLIC 246

SLIC 248

C	-----	SLIC	250
C	DIVIDE SLIDING MASS AT EACH GROUND SURFACE	SLIC	252
C	END POINT WITHIN SLIDING MASS	SLIC	254
C	-----	SLIC	256
C	DO 2 I=1,NTOP	SLIC	258
	IF(BNDS(I,3).LE.SURF(1,1))GO TO 2	SLIC	260
	IF(BNDS(I,3).GE.SURF(NSURF,1))GO TO 28	SLIC	262
	K=K+1	SLIC	264
	X(K)=BNDS(I,3)	SLIC	266
	2 CONTINUE	SLIC	268
	28 IF(NTOP.EQ.NBND)GO TO 33	SLIC	270
C	-----	SLIC	272
C	DIVIDE SLIDING MASS AT EACH SUBSURFACE BOUNDARY	SLIC	274
C	END POINT WITHIN THE SLIDING MASS.	SLIC	276
C	-----	SLIC	278
C	NTOP1=NTOP+1	SLIC	280
	DO 27 I=NTOP1,NBND	SLIC	282
	IF(BNDS(I,3).LE.SURF(1,1))GO TO 27	SLIC	284
	IF(BNDS(I,1).GE.SURF(NSURF,1))GO TO 27	SLIC	286
	IF(BNDS(I-1,3).EQ.BNDS(I,1))GO TO 29	SLIC	288
	IF(BNDS(I,1).LE.SURF(1,1))GO TO 29	SLIC	290
	II=1	SLIC	292
	GO TO 30	SLIC	294
	29 IF(BNDS(I,3).GE.SURF(NSURF,1))GO TO 27	SLIC	296
	II=3	SLIC	298
	30 DO 12 J=2,NSURF	SLIC	300
	32 IF(BNDS(I,II).GT.SURF(J,1))GO TO 12	SLIC	302
	CALL INTSC2(SURF(J-1,1),SURF(J-1,2),SURF(J,1),SURF(J,2),0.,0.,0.,	SLIC	304
	1 0.,BNDS(I,II),Y,INTS)	SLIC	306
	IF(BNDS(I,II+1).LT.Y)GO TO 31	SLIC	308
	K=K+1	SLIC	310
	X(K)=BNDS(I,II)	SLIC	312
	31 IF(II.EQ.3)GO TO 27	SLIC	314
	II=3	SLIC	316
	GO TO 32	SLIC	318
	12 CONTINUE	SLIC	320
	27 CONTINUE	SLIC	322
	33 IF(IWAT.EQ.0)GO TO 5	SLIC	324
C	-----	SLIC	326
C	DIVIDE SLIDING MASS AT EACH	SLIC	328
C	POINT DEFINING THE WATER SURFACE	SLIC	330
C	-----	SLIC	332
C	DO 35 N=1,NPZ	SLIC	334
	JS=2	SLIC	336
	NN=NPZ(N)	SLIC	338
	DO 4 I=1,NN	SLIC	340
	IF(XPIEZ(N,I).LT.SURF(1,1))GO TO 4	SLIC	342
	IF(XPIEZ(N,I).GE.SURF(NSURF,1))GO TO 4	SLIC	344
	DO 13 J=JS,NSURF	SLIC	346
	JTN=J	SLIC	348
	IF(XPIEZ(N,I).GT.SURF(J,1))GO TO 13	SLIC	350
	CALL INTSC2(SURF(J-1,1),SURF(J-1,2),SURF(J,1),SURF(J,2),0.,0.,0.,	SLIC	352
	1 0.,XPIEZ(N,I),Y,INTS)	SLIC	353
	IF(YPIEZ(N,I).LT.Y)GO TO 14	SLIC	354
	K=K+1	SLIC	356
	X(K)=XPIEZ(N,I)	SLIC	358
	GO TO 14	SLIC	360
	13 CONTINUE	SLIC	362
	14 JS=JTN	SLIC	364
	4 CONTINUE	SLIC	366
	35 CONTINUE	SLIC	368
	5 IF(ISURC.EQ.0)GO TO 6	SLIC	370
C		SLIC	372
		SLIC	374
		SLIC	376

C	-----	SLIC 378
C	DIVIDE SLIDING MASS AT EACH POINT	SLIC 380
C	DEFINING A CHANGE IN SURFACE LOADING	SLIC 382
C	-----	SLIC 384
C		SLIC 386
	DO 7 I=1, NSURC	SLIC 388
	IF(SURC(I,2).LE.SURF(1,1))GO TO 7	SLIC 390
	IF(SURC(I,1).GE.SURF(NSURF,1))GO TO 7	SLIC 392
	IF(SURC(I,1).GT.SURF(1,1))GO TO 8	SLIC 394
	K=K+1	SLIC 396
	X(K)=SURC(I,2)	SLIC 399
	GO TO 7	SLIC 400
	8 IF(SURC(I,2).LT.SURF(NSURF,1))GO TO 9	SLIC 402
	K=K+1	SLIC 404
	X(K)=SURC(I,1)	SLIC 406
	GO TO 7	SLIC 408
	9 K=K+1	SLIC 410
	X(K)=SURC(I,1)	SLIC 412
	K=K+1	SLIC 414
	X(K)=SURC(I,2)	SLIC 416
	7 CONTINUE	SLIC 418
	6 IF(NTOP.EQ.NBND)GO TO 20	SLIC 420
	NTOP1=NTOP+1	SLIC 422
	DO 18 I=NTOP1, NBND	SLIC 424
C	-----	SLIC 426
C	WHEN TRIAL FAILURE SURFACE INTERSECTS A SUBSURFACE	SLIC 428
C	BOUNDARY, DIVIDE SLIDING MASS AT THAT POINT	SLIC 430
C	-----	SLIC 432
C		SLIC 434
	DO 10 J=2, NSURF	SLIC 436
	CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),SURF(J-1,1),	SLIC 438
	1 SURF(J-1,2),SURF(J,1),SURF(J,2),XINT,YINT,INTS)	SLIC 440
	IF(INTS.NE.1)GO TO 10	SLIC 442
	K=K+1	SLIC 444
	X(K)=XINT	SLIC 446
	10 CONTINUE	SLIC 448
	IF(IWAT.EQ.0)GO TO 18	SLIC 450
C	-----	SLIC 452
C	WHEN FREE WATER SURFACE INTERSECTS A SUBSURFACE	SLIC 454
C	BOUNDARY, DIVIDE SLIDING MASS AT THAT POINT	SLIC 456
C	-----	SLIC 458
C		SLIC 460
	DO 36 N=1, NPZ	SLIC 462
	NN=NPIEZ(N)	SLIC 463
	DO 19 J=2, NN	SLIC 464
	IF(XPIEZ(N,J).LE.SURF(1,1).OR.XPIEZ(N,J-1).GE.SURF(NSURF,1))	SLIC 465
	1 GO TO 19	SLIC 466
	CALL INTSCT(BNDS(I,1),BNDS(I,2),BNDS(I,3),BNDS(I,4),XPIEZ(N,J-1),	SLIC 468
	1 YPIEZ(N,J-1),XPIEZ(N,J),YPIEZ(N,J),XINT,YINT,INTS)	SLIC 469
	IF(INTS.NE.1)GO TO 19	SLIC 470
	IF(XINT.LE.SURF(1,1).OR.XINT.GE.SURF(NSURF,1))GO TO 19	SLIC 472
	DO 21 II=2, NSURF	SLIC 474
	IF(XINT.GT.SURF(II,1))GO TO 21	SLIC 476
	CALL INTSC2(SURF(II-1,1),SURF(II-1,2),SURF(II,1),SURF(II,2),0.,0.,	SLIC 478
	1 0.,0.,XINT,Y,INTS)	SLIC 480
	IF(YINT.LT.Y)GO TO 19	SLIC 482
	K=K+1	SLIC 484
	X(K)=XINT	SLIC 486
	GO TO 19	SLIC 488
	21 CONTINUE	SLIC 490
	19 CONTINUE	SLIC 492
	36 CONTINUE	SLIC 494
	18 CONTINUE	SLIC 496
	20 IF(IWAT.EQ.0)GO TO 22	SLIC 497
C		SLIC 498
		SLIC 500
		SLIC 502

C	-----	SLIC 504
C	WHEN WATER SURFACE INTERSECTS THE TRIAL FAILURE	SLIC 506
C	SURFACE, DIVIDE SLIDING MASS AT THAT POINT	SLIC 508
C	-----	SLIC 510
C	DO 37 N=1,NPZ	SLIC 512
	JW=2	SLIC 513 -
	JW=2	SLIC 514
	NN=NPIEZ(N)	SLIC 516
	DO 23 I=2,NSURF	SLIC 517 -
	DO 24 J=JW,NN	SLIC 518
	IF(XPIEZ(N,J).LE.SURF(I-1,1))GO TO 25	SLIC 520
	IF(XPIEZ(N,J-1).GE.SURF(I,1))GO TO 26	SLIC 522
	CALL INTSCT(XPIEZ(N,J-1),YPIEZ(N,J-1),XPIEZ(N,J),YPIEZ(N,J),	SLIC 524
	1 SURF(I-1,1),SURF(I-1,2),SURF(I,1),SURF(I,2),XINT,YINT,INTS)	SLIC 526
	IF(INTS.EQ.0)GO TO 24	SLIC 528
	K=K+1	SLIC 530
	X(K)=XINT	SLIC 532
	25 JW=J	SLIC 534
	24 CONTINUE	SLIC 536
	26 JW=JW	SLIC 538
	23 CONTINUE	SLIC 540
	37 CONTINUE	SLIC 542
C	-----	SLIC 543 -
C	ORDER SLICE DIVISIONS FROM LEFT TO RIGHT	SLIC 544
C	-----	SLIC 546
C	22 DO 11 I=1,K	SLIC 548
	DO 11 J=2,K	SLIC 550
	IF(X(J-1).LE.X(J))GO TO 11	SLIC 552
	A=X(J)	SLIC 554
	X(J)=X(J-1)	SLIC 556
	X(J-1)=A	SLIC 558
	11 CONTINUE	SLIC 560
	J=2	SLIC 562
		SLIC 564
		SLIC 566
		SLIC 568
		SLIC 570
C	-----	SLIC 572
C	CHECK FOR AND ELIMINATE DUPLICATE DIVISIONS	SLIC 574
C	-----	SLIC 576
C	17 IF(ABS(X(J-1)-X(J)).GE.TOL)GO TO 15	SLIC 578
	DO 16 I=J,K	SLIC 580
	X(I-1)=X(I)	SLIC 582
	16 CONTINUE	SLIC 584
	K=K-1	SLIC 586
	GO TO 34	SLIC 588
	15 J=J+1	SLIC 590
	34 IF(J.LE.K)GO TO 17	SLIC 592
	NSLICE=K-1	SLIC 594
	RETURN	SLIC 596
	END	SLIC 598
		SLIC 600

	SUBROUTINE WEIGHT	WGHT	2
	-----	WGHT	4
	-----	WGHT	6
	SUBROUTINE WEIGHT	WGHT	8
	-----	WGHT	10
	-----	WGHT	12
	-----	WGHT	14
FUNCTIONS -		WGHT	16
	DETERMINES THE INCLINATION OF THE TOP AND BOTTOM OF EACH SLICE.	WGHT	18
		WGHT	20
		WGHT	22
		WGHT	24
	DETERMINES THE RESULTANT WATER FORCES AT THE TOP AND BASE OF EACH SLICE IF PRESENT.	WGHT	26
		WGHT	28
		WGHT	30
	DETERMINES THE TOTAL WEIGHT OF EACH SLICE AND THE SOIL TYPE AT THE BASE OF EACH SLICE.	WGHT	32
		WGHT	34
		WGHT	36
	DETERMINES THE SURCHARGE FORCE, IF ANY, AT THE TOP OF EACH SLICE.	WGHT	38
		WGHT	40
		WGHT	42
	-----	WGHT	44
	-----	WGHT	46
DEFINITIONS -		WGHT	48
		WGHT	50
ALPHA	ARRAY CONTAINING VALUES OF THE ANGLES OF THE BASE OF EACH SLICE.	WGHT	52
		WGHT	54
		WGHT	56
ATAN	STANDARD FUNCTION THAT CALCULATES THE ARCTANGENT OF A NUMBER.	WGHT	58
		WGHT	60
		WGHT	62
BETA	ARRAY CONTAINING VALUES OF THE ANGLE OF THE TOP OF EACH SLICE.	WGHT	64
		WGHT	66
		WGHT	68
BNDS	AN ARRAY CONTAINING THE COORDINATES OF THE END POINTS DEFINING THE GROUND SURFACE AND SUBSURFACE BOUNDARIES.	WGHT	70
		WGHT	72
		WGHT	74
		WGHT	76
CA	COSINE OF ANGLE ALPHA.	WGHT	78
		WGHT	80
CAUT	CAVITATION PRESSURE.	WGHT	82
		WGHT	84
COS	STANDARD FUNCTION THAT CALCULATES THE COSINE OF AN ANGLE.	WGHT	86
		WGHT	88
		WGHT	90
CU	ARRAY CONTAINING VALUES OF CONSTANT PORE PRESSURE FOR EACH SOIL TYPE.	WGHT	92
		WGHT	94
		WGHT	96
DELT	ARRAY CONTAINING VALUES OF INCLINATION OF THE BOUNDARY LOAD FOR EACH SLICE.	WGHT	98
		WGHT	100
		WGHT	102
DELTA	ARRAY CONTAINING VALUES OF LOAD DIRECTION FOR EACH BOUNDARY LOAD SPECIFIED.	WGHT	104
		WGHT	106
		WGHT	108
DX	ARRAY CONTAINING VALUES OF THE WIDTH OF EACH SLICE.	WGHT	110
		WGHT	112
I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	WGHT	114
		WGHT	116
INTS	CONTROL CODE WHICH SIGNALS WHETHER AN INTERSECTION HAS OCCURRED OR NOT.	WGHT	118
		WGHT	120
		WGHT	122
INTSC2	ENTRY POINT INTO SUBROUTINE INTSCT WHICH DETERMINES WHETHER OR NOT A LINE SEGMENT AND A VERTICAL LINE INTERSECT AND CALCULATES THE COORDINATES OF THE INTERSECTION.	WGHT	124
		WGHT	126
		WGHT	128
		WGHT	130
		WGHT	132

C	ISEARC	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	WGHT 134
C		AFFECTED BY THE SEARCHING SUBROUTINE RANDOM.	WGHT 136
C			WGHT 138
C	ISURC	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	WGHT 140
C		HANDLING THE SURCHARGE LOADS SPECIFIED BY SUBROUTINE	WGHT 142
C		LOADS.	WGHT 144
C			WGHT 146
C	ITP	ARRAY CONTAINING SOIL TYPE INDICES FOR EACH BOUNDARY.	WGHT 148
C			WGHT 150
C	IWAT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	WGHT 152
C		HANDLING THE WATER SURFACE SPECIFIED BY SUBROUTINE	WGHT 154
C		WATER.	WGHT 156
C			WGHT 158
C	J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	WGHT 160
C			WGHT 162
C	JB	SUBSCRIPT OF LAST POINT, DEFINING THE TRIAL FAILURE	WGHT 164
C		SURFACE, TO BE USED FOR DETERMINING THE INTERSECTION	WGHT 166
C		OF THE CENTERLINE OF A SLICE WITH THE FAILURE	WGHT 168
C		SURFACE.	WGHT 170
C			WGHT 172
C	JJ	VARIABLE USED TO TEMPORARY STORE AN ARRAY VARIABLE	AUG76EBO -
C		FOR A DO STATEMENT.	AUG76EBO -
C			AUG76EBO -
C	JS	SUBSCRIPT OF LAST BOUNDARY LOAD USED FOR DETERMINING	WGHT 174
C		LOAD DISTRIBUTION TO A SLICE.	WGHT 176
C			WGHT 178
C	JT	SUBSCRIPT OF LAST BOUNDARY, DEFINING THE GROUND	WGHT 180
C		SURFACE, TO BE USED FOR DETERMINING THE INTERSECTION	WGHT 182
C		OF THE CENTERLINE OF A SLICE WITH THE GROUND SURFACE.	WGHT 184
C			WGHT 186
C	JTN	VARIABLE USED TO TEMPORARY STORE THE VALUE OF AN	WGHT 188
C		INDEX VARIABLE USED FOR ARRAY SUBSCRIPTING.	WGHT 190
C			WGHT 192
C	JW	ARRAY CONTAINING SUBSCRIPT OF LAST POINT DEFINING	WGHT 194
C		EACH PIEZOMETRIC SURFACE, USED TO DETERMINE THE	WGHT 196
C		INTERSECTION OF THE CENTERLINE OF A SLICE WITH	WGHT 198
C		THE PIEZOMETRIC SURFACE.	WGHT 200
C			WGHT 202
C	K	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	WGHT 204
C			WGHT 206
C	K1	K-1	AUG76EBO -
C			AUG76EBO -
C	KCOEF	HORIZONTAL EARTHQUAKE COEFFICIENT.	WGHT 208
C			WGHT 210
C	LOAD	ARRAY CONTAINING VALUES OF UNIFORM LOAD INTENSITY	WGHT 212
C		FOR EACH BOUNDARY LOAD SPECIFIED.	WGHT 214
C			WGHT 216
C	N	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	WGHT 218
C			WGHT 220
C	NBND	TOTAL NUMBER OF PROFILE BOUNDARIES.	WGHT 222
C			WGHT 224
C	NN	VARIABLE USED TO TEMPORARY STORE AN ARRAY VARIABLE	AUG76EBO
C		FOR A DO STATEMENT.	AUG76EBO
C			AUG76EBO
C	NP	ARRAY CONTAINING NUMBER OF PIEZOMETRIC SURFACE	WGHT 226
C		FOR EACH SOIL TYPE.	WGHT 228
C			WGHT 230
C	NPIEZ	NUMBER OF POINTS DEFINING THE WATER SURFACE.	WGHT 232
C			WGHT 234
C	NPZ	NUMBER OF PIEZOMETRIC SURFACES DEFINED.	WGHT 236 -
C			WGHT 238 -
C	NSLICE	NUMBER OF SLICES WHICH THE SLIDING MASS IS DIVIDED.	WGHT 240
C			WGHT 242
C	NSOIL	NUMBER OF SOIL TYPES SPECIFIED.	NOV76EBO -
C			NOV76EBO -
C	NSURC	NUMBER OF BOUNDARY LOADS SPECIFIED.	WGHT 244
C			WGHT 246

NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	WGHT 248
NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	WGHT 250
NTOP1	NTOP + 1	WGHT 252
P	ARRAY CONTAINING VALUES OF THE BOUNDARY LOAD APPLIED TO EACH SLICE.	WGHT 254
		WGHT 256
		WGHT 258
PLT4	ENTRY OF SUBROUTINE PLTN WHICH STORES THE PROFILE GEOMETRY AND, IF APPLICABLE, THE WATER SURFACE, SURFACE GENERATION BOUNDARIES, SURCHARGE BOUNDARY LOADS AND A SPECIFIED TRIAL FAILURE SURFACE. PLT4 THEN PRINTS THE ENTIRE ARRAY AND ITS AXIS LABELS.	WGHT 260
		WGHT 262
		WGHT 264
		NOU76E80
		NOU76E80
		NOU76E80
		NOU76E80
		NOU76E80
QUIT	SUBROUTINE THAT DISPLAYS A TERMINATION MESSAGE, TERMINATES PLOTTING, AND TERMINATES EXECUTION OF THE PROGRAM.	WGHT 266
		WGHT 268
		WGHT 270
		WGHT 272
RU	ARRAY CONTAINING VALUES OF THE PORE PRESSURE PARAMETER FOR EACH SOIL TYPE.	WGHT 274
		WGHT 276
		WGHT 278
SIN	STANDARD FUNCTION THAT CALCULATES THE SINE OF AN ANGLE.	WGHT 280
		WGHT 282
		WGHT 284
SLTP	ARRAY CONTAINING INDICES OF THE SOIL TYPE AT THE BASE OF EACH SLICE.	WGHT 286
		WGHT 288
		WGHT 290
SOIL	ARRAY CONTAINING INDICES OF THE SOIL TYPE OF EACH SUBSECTION WITHIN A SLICE.	WGHT 292
		WGHT 294
		WGHT 296
SOILTP	VARIABLE CONTAINING SOIL TYPE AT THE BASE OF A SLICE.	WGHT 298
		WGHT 300
SOILWT	SUBROUTINE THAT CALCULATES THE TOTAL WEIGHT OF A SLICE SUBSECTION.	WGHT 302
		WGHT 304
		WGHT 306
SURC	ARRAY CONTAINING X COORDINATES OF THE END POINTS DEFINING THE EXTENT OF LOADING.	WGHT 308
		WGHT 310
		WGHT 312
SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS DEFINING A TRIAL FAILURE SURFACE.	WGHT 314
		WGHT 316
		WGHT 318
UALPHA	ARRAY CONTAINING VALUES OF THE HYDROSTATIC FORCE ACTING AT THE BASE OF EACH SLICE.	WGHT 320
		WGHT 322
		WGHT 324
UBETA	ARRAY CONTAINING VALUES OF THE HYDROSTATIC FORCE ACTING AT THE TOP OF EACH SLICE.	WGHT 326
		WGHT 328
		WGHT 330
UWAT	UNITWEIGHT OF WATER. IF 0. IS SPECIFIED 62.4 PCF IS ASSUMED.	WGHT 332 -
		WGHT 334 -
		WGHT 336 -
UKCOEF	VERTICAL EARTHQUAKE COEFFICIENT.	WGHT 338
		WGHT 340
WT	WEIGHT OF A SLICE SUBSECTION.	WGHT 342
		WGHT 344
WTT	ARRAY CONTAINING VALUES OF THE WEIGHT OF EACH SLICE.	WGHT 346
		WGHT 348
X	ARRAY CONTAINING THE POSITIONS OF THE CENTERLINE FOR EACH SLICE.	WGHT 350
		WGHT 352
		WGHT 354
XPIEZ	ARRAY CONTAINING X COORDINATES OF POINTS DEFINING WATER SURFACE.	WGHT 356
		WGHT 358
		WGHT 360
Y	ARRAY CONTAINING Y COORDINATES OF THE PIEZOMETRIC SURFACES FOR A SLICE.	WGHT 362 -
		WGHT 364 -
		WGHT 366 -
YB	Y COORDINATE OF THE SLICE BASE.	WGHT 368
		WGHT 370

C	YI	ARRAY CONTAINING THE Y COORDINATES OF INTERMEDIATE	WGHT 372
C		POINTS DIVIDING A SLICE INTO SUBSECTIONS.	WGHT 374
C	YPIEZ	ARRAY CONTAINING Y COORDINATES OF POINTS DEFINING	WGHT 376
C		WATER SURFACE.	WGHT 378
C	YT	Y COORDINATE OF THE TOP OF A SLICE.	WGHT 380
C			WGHT 382
C	YW	ARRAY CONTAINING Y COORDINATE OF THE PIEZOMETRIC	WGHT 384
C		SURFACE FOR EACH SOIL TYPE OF A SLICE.	WGHT 386
C			WGHT 388
C			WGHT 390
C			WGHT 392
C			WGHT 394
C			WGHT 396
C			WGHT 398
	COMMON /BLK01/	IANG1,IBLK,IEXIT,ICIRC,ILIMIT,IPL0T,IREAD,ISEARC,	WGHT 400
	1	IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL	WGHT 402 -
	COMMON /BLK02/	BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND,	WGHT 404
	1	NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20)	WGHT 406 -
	COMMON /BLK03/	UWAT,NPZ,NPIEZ(10),XPIEZ(10,40),YPIEZ(10,40)	WGHT 408 -
	COMMON /BLK04/	DELTA(10),LOAD(10),NSURC,SURC(10,2),YSURC(10,2)	WGHT 410
	COMMON /BLK05/	NSURF,SURF(100,2)	WGHT 412
	COMMON /BLK08/	NSLICE,X(300)	WGHT 414
	COMMON /BLK09/	ALPHA(200),BETA(200),DX(200),SLTP(200),UALPHA(200),	WGHT 416
	1	UBETA(200),WTT(200)	WGHT 418 -
	COMMON /BLK10/	DELT(200),P(200)	WGHT 420
	COMMON /BLK11/	CAUT,KCOEF,UKCOEF	WGHT 422
	DIMENSION	YI(20),YW(20),JW(10),Y(10)	WGHT 424 -
	REAL	KCOEF,LOAD	WGHT 426
	INTEGER	SOIL(20),SOILTP,SLTP	WGHT 428
	JT=1		WGHT 430
	JB=2		WGHT 432
	JS=1		WGHT 434
	DO 13 J=1,NPZ		WGHT 436 -
	JW(J)=2		WGHT 438 -
	13 CONTINUE		WGHT 440 -
	NTOP1=NTOP+1		WGHT 442
C			WGHT 444
C			WGHT 446
C			WGHT 448
C			WGHT 450
	DO 2 I=1,NSLICE		WGHT 452
	DX(I)=X(I+1)-X(I)		WGHT 454
	X(I)=(X(I)+X(I+1))/2.		WGHT 456
	DO 16 J=1,NSOIL		WGHT 458
	YW(J)=0.		WGHT 460 -
	16 CONTINUE		WGHT 462 -
			WGHT 464 -
			WGHT 466
			WGHT 468
			WGHT 470
			WGHT 472
			WGHT 474
	DO 3 J=JT,NTOP		WGHT 476
	JTN=J		WGHT 478 -
	CALL INTSC2(BNDS(J,1),BNDS(J,2),BNDS(J,3),BNDS(J,4),0.,0.,0.,0.,		WGHT 480
	1	X(I),YT,INTS)	WGHT 482
	IF(INTS.EQ.1)GO TO 5		WGHT 484
	3 CONTINUE		WGHT 486
	5 JT=JTN		WGHT 488 -
	J=JTN		WGHT 490 -
	BETA(I)=ATAN((BNDS(J,2)-BNDS(J,4))/(BNDS(J,1)-BNDS(J,3)))		WGHT 492
C			WGHT 494

C	-----	WGHT	496
C	DETERMINE BASE OF SLICE AND ITS INCLINATION	WGHT	498
C	-----	WGHT	500
C		WGHT	502
	DO 4 J=JB,NSURF	WGHT	504
	JTN=J	WGHT	506
	CALL INTSC2(SURF(J-1,1),SURF(J-1,2),SURF(J,1),SURF(J,2),0.,0.,0.,	WGHT	508
	1 0.,X(I),YB,INTS)	WGHT	510
	IF(INTS.EQ.1)GO TO 6	WGHT	512
	4 CONTINUE	WGHT	514
	6 JB=JTN	WGHT	516
	J=JTN	WGHT	518
	ALPHA(I)=ATAN((SURF(J-1,2)-SURF(J,2))/(SURF(J-1,1)-SURF(J,1)))	WGHT	520
	CA=COS(ALPHA(I))	WGHT	522
C	-----	WGHT	524
C		WGHT	526
C	CHECK IF TRIAL SURFACE IS BELOW THE GROUND SURFACE	WGHT	528
C	-----	WGHT	530
C		WGHT	532
	IF(YT.GT.YB)GO TO 18	WGHT	534
	WRITE(6,101)	WGHT	536
101	FORMAT(/,	WGHT	538
	110X,*****//	WGHT	540
	110X,***** INPUT ERROR - TRIAL FAILURE SURFACE *****//	WGHT	542
	110X,***** EXTENDS ABOVE THE *****//	WGHT	544
	110X,***** GROUND SURFACE *****//	WGHT	546
	110X,*****//)	WGHT	548
	KEY= 72	KEY	72
	WRITE(6,72)KEY,I,X(I),YT,YB,DX(I)	WRITEKEY	
72	FORMAT(/,XXX KEY=,I4,XXX ,I10,5F15.5)	KEYFORMT	
	IF(ISEARC.EQ.0)GO TO 22	WGHT	550
	WRITE(6,104)NSURF	WGHT	552
104	FORMAT(/,,,	WGHT	554
	110X,THE TRIAL FAILURE SURFACE IN QUESTION IS DEFINED,/,	WGHT	556
	110X,BY THE FOLLOWING,I3,	WGHT	558
	1 COORDINATE POINTS,/,/,	WGHT	560
	1 12X,POINT,6X,X-SURF,6X,Y-SURF,/,	WGHT	562
	1 13X,NO,8X,(FT),8X,(FT),/,	WGHT	564
	WRITE(6,105)(J,SURF(J,1),SURF(J,2),J=1,NSURF)	WGHT	566
105	FORMAT(12X,I3,2X,2F12.2)	WGHT	568
	22 CALL PLT4	WGHT	570
	CALL QUIT	WGHT	572
	18 K=1	WGHT	574
	SOIL(1)=ITP(JT)	WGHT	576
	IF(NTOP.EQ.NBND)GO TO 12	WGHT	578
C	-----	WGHT	580
C		WGHT	582
C	DETERMINE POSITIONS OF SOIL TYPE INTERFACES, IF ANY	WGHT	584
C	AND DETERMINE SOIL TYPE AT BASE OF SLICE.	WGHT	586
C	-----	WGHT	588
C		WGHT	590
	DO 9 J=NTOP1,NBND	WGHT	592
	CALL INTSC2(BNDS(J,1),BNDS(J,2),BNDS(J,3),BNDS(J,4),0.,0.,0.,0.,	WGHT	594
	1 X(I),YI(K),INTS)	WGHT	596
	IF(INTS.NE.1)GO TO 9	WGHT	598
	IF(YI(K).LE.YB)GO TO 12	WGHT	600
	K=K+1	WGHT	602
	SOIL(K)=ITP(J)	WGHT	604
9	CONTINUE	WGHT	606
12	SLTP(I)=SOIL(K)	WGHT	608
	SOILTP=SLTP(I)	WGHT	610
	UALPHA(I)=0.	WGHT	612
	UBETA(I)=0.	WGHT	614
	IF(IWAT.EQ.0)GO TO 10	WGHT	616
C		WGHT	618

C	-----	WGHT 620
C	DETERMINE POSITION OF WATER TABLE FOR ALL SOILS IF PRESENT	WGHT 622
C	-----	WGHT 624
C	DO 25 N=1, NPZ	WGHT 626
	JJ=JW(N)	WGHT 627
	NN=NPIEZ(N)	WGHT 628
	DO 7 J=JJ, NN	WGHT 629
	JTN=J	WGHT 630
	CALL INTSC2(XPIEZ(N, J-1), YPIEZ(N, J-1), XPIEZ(N, J), YPIEZ(N, J),	WGHT 632
	1 0., 0., 0., 0., X(I), Y(N), INTS)	WGHT 634
	IF(INTS.EQ.1)GO TO 8	WGHT 636
	7 CONTINUE	WGHT 638
	8 JW(N)=JTN	WGHT 640
	25 CONTINUE	WGHT 642
	DO 25 J=1, K	WGHT 644
	YW(J)=Y(NP(SOIL(J)))	WGHT 646
	26 CONTINUE	WGHT 648
	IF(YW(K).GT.YB)UALPHA(I)=(YW(K)-YB)*UWAT*DX(I)/CA	WGHT 650
	IF(YW(1).LT.YT)GO TO 10	WGHT 652
	-----	WGHT 654
C	-----	WGHT 656
C	CALCULATE RESULTANT WATER FORCE AT TOP OF SLICE, IF PRESENT	WGHT 658
C	-----	WGHT 660
C	UBETA(I)=DX(I)*(YW(1)-YT)*UWAT/COS(BETA(I))	WGHT 662
C	-----	WGHT 664
C	-----	WGHT 666
C	CALCULATE SLICE WEIGHT	WGHT 668
C	-----	WGHT 670
C	-----	WGHT 672
C	10 IF(K.EQ.1)GO TO 14	WGHT 674
	CALL SOILWT(YT, YI(1), ITP(JT), YW(1), WTT(I), I)	WGHT 676
	GO TO 15	WGHT 678
	14 CALL SOILWT(YT, YB, ITP(JT), YW(1), WTT(I), I)	WGHT 680
	GO TO 21	WGHT 682
	15 IF(K.EQ.2)GO TO 17	WGHT 684
	K1=K-1	WGHT 686
	DO 11 J=2, K1	WGHT 688
	CALL SOILWT(YI(J-1), YI(J), SOIL(J), YW(J), WT, I)	WGHT 689
	WTT(I)=WTT(I)+WT	WGHT 690
	11 CONTINUE	WGHT 692
	17 CALL SOILWT(YI(K-1), YB, SOIL(K), YW(K), WT, I)	WGHT 694
	WTT(I)=WTT(I)+WT	WGHT 696
	21 IF(YW(K).LT.YB.AND.RU(SOILT).EQ.0..AND.CU(SOILT).EQ.0.)GO TO 23	WGHT 698
	IF(RU(SOILT).EQ.0..AND.CU(SOILT).EQ.0.)GO TO 24	WGHT 700
	UALPHA(I)=UALPHA(I)+(WTT(I)*RU(SOILT)+DX(I)*CU(SOILT))/CA	WGHT 702
	24 IF(KCOEF.EQ.0..AND.UKCOEF.EQ.0.)GO TO 23	WGHT 704
	UALPHA(I)=UALPHA(I)-WTT(I)*(KCOEF*SIN(ALPHA(I))+UKCOEF*CA)	WGHT 706
	IF(UALPHA(I).LT.CAUT)UALPHA(I)=CAUT	WGHT 708
	23 P(I)=0.	WGHT 710
	DELT(I)=0.	WGHT 712
	IF(ISURC.EQ.0)GO TO 2	WGHT 714
	-----	WGHT 716
C	-----	WGHT 718
C	ASSIGN BOUNDARY LOAD TO SLICE, IF ANY	WGHT 720
C	-----	WGHT 722
C	-----	WGHT 724
C	DO 19 J=JS, NSURC	WGHT 726
	JTN=J	WGHT 728
	IF(X(I).LT.SURC(J,1))GO TO 20	WGHT 730
	IF(X(I).GT.SURC(J,2))GO TO 19	WGHT 732
	DELT(I)=DELT(J)	WGHT 734
	P(I)=LOAD(J)*DX(I)	WGHT 736
	GO TO 20	WGHT 738
	19 CONTINUE	WGHT 740
	20 JS=JTN	WGHT 742
	2 CONTINUE	WGHT 744
	RETURN	WGHT 746
	END	WGHT 748
		WGHT 750
		WGHT 752


```

C SUBROUTINE SOILWT(UB, LB, ST, YW, WT, I) SLWT 2
C ----- SLWT 4
C ----- SLWT 6
C SUBROUTINE SOILWT SLWT 8
C ----- SLWT 10
C ----- SLWT 12
C SLWT 14
C FUNCTION - SLWT 16
C SLWT 18
C CALCULATES THE TOTAL WEIGHT OF A SLICE SUBSECTION. SLWT 20
C SLWT 22
C ----- SLWT 24
C SLWT 26
C DEFINITIONS - SLWT 28
C SLWT 30
C A TOTAL AREA OF SLICE SUBSECTION. SLWT 32
C SLWT 34
C AB AREA OF SLICE SUBSECTION BELOW FREEWATER SURFACE. SLWT 36
C SLWT 38
C AT AREA OF SLICE SUBSECTION ABOVE FREEWATER SURFACE. SLWT 40
C SLWT 42
C DX ARRAY CONTAINING VALUES OF THE WIDTH OF EACH SLICE. SLWT 44
C SLWT 46
C GAMMA ARRAY CONTAINING VALUES OF THE TOTAL UNIT WEIGHT FOR SLWT 48
C EACH SOIL TYPE. SLWT 50
C SLWT 52
C GSAT ARRAY CONTAINING VALUES OF THE SATURATED UNIT WEIGHT SLWT 54
C FOR EACH SOIL TYPE. SLWT 56
C SLWT 58
C I INDEX VARIABLE FOR ARRAY SUBSCRIPTING. SLWT 60
C SLWT 62
C LB Y COORDINATE OF SLICE BASE. SLWT 64
C SLWT 66
C ST SOIL TYPE OF SUBSECTION. SLWT 68
C SLWT 70
C UB Y COORDINATE OF TOP OF SLICE. SLWT 72
C SLWT 74
C WT WEIGHT OF SLICE SUBSECTION. SLWT 76
C SLWT 78
C YW Y COORDINATE OF PIEZOMETRIC SURFACES FOR SLICE. SLWT 80
C SLWT 82
C ----- SLWT 84
C ----- SLWT 86
C SLWT 88
C COMMON /BLK02/BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND, SLWT 90
C 1 NSOIL,NTDP,PHI(20),RU(20),CU(20),NP(20) SLWT 92
C COMMON /BLK09/ALPHA(200),BETA(200),DX(200),SLTP(200),UALPHA(200), SLWT 94
C 1 UBETA(200),WTT(200) SLWT 96
C INTEGER ST SLWT 98
C REAL LB SLWT 100
C SLWT 102
C SLWT 104
C CHECK FOR WATER TABLE POSITION, IF PRESENT SLWT 106
C ----- SLWT 108
C SLWT 110
C IF(YW.LT.UB.AND.YW.GT.LB)GO TO 2 SLWT 112
C SLWT 114
C ----- SLWT 116
C CALCULATE DRY WEIGHT OF SLICE SUBAREA SLWT 118
C ----- SLWT 120
C SLWT 122
C A=DX(I)*(UB-LB) SLWT 124
C WT=A*GAMMA(ST) SLWT 126
C SLWT 128

```


C	-----	SLWT 130
C	CALCULATE SATURATED WEIGHT OF SLICE SUBAREA, IF SATURATED	SLWT 132
C	-----	SLWT 134
C	IF(YW.LT.UB)RETURN	SLWT 136
	WT=A*GSAT(ST)	SLWT 138
	RETURN	SLWT 140
C		SLWT 142
C	-----	SLWT 144
C		SLWT 146
C	CALCULATE WEIGHT OF PARTIALLY SATURATED SLICE SUBAREA, IF APPLICABLE	SLWT 148
C	-----	SLWT 150
C		SLWT 152
	2 AT=DX(I)*(UB-YW)	SLWT 154
	AB=DX(I)*(YW-LB)	SLWT 156
	WT=AT*GAMMA(ST)+AB*GSAT(ST)	SLWT 158
	RETURN	SLWT 160
	END	SLWT 162

C	DX	ARRAY CONTAINING VALUES OF THE WIDTH OF EACH SLICE.	FCTR 138
C			FCTR 140
C	F	CRITICAL LOCAL FACTOR OF SAFETY USED TO DETECT A	FCTR 142
C		NEGATIVE NORMAL FORCE ACTING ON A SLICE.	FCTR 144
C			FCTR 146
C	FNEW	NEW FACTOR OF SAFETY USED DURING ITERATIVE PROCESS.	FCTR 148
C			FCTR 150
C	FOLD	OLD FACTOR OF SAFETY USED DURING ITERATIVE PROCESS.	FCTR 152 -
C			FCTR 154
C	FS	FACTOR OF SAFETY. . .	FCTR 156
C			FCTR 158
C	F1	LARGEST CRITICAL LOCAL FACTOR OF SAFETY USED TO	FCTR 160
C		DETECT A NEGATIVE NORMAL FORCE ACTING ON A SLICE.	FCTR 162
C			FCTR 164
C	I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	FCTR 166
C			FCTR 168
C	ISEARC	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	FCTR 170
C		AFFECTED BY THE SEARCHING SUBROUTINE RANDOM.	FCTR 172
C			FCTR 174
C	ISTR	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	FCTR 176
C		HANDLING THE ANISOTROPIC SOIL STRENGTH DATA.	FCTR 178
C			FCTR 180
C	ITPA	ARRAY CONTAINING THE INDICES OF ANISOTROPIC SOIL	FCTR 182
C		TYPES.	FCTR 184
C			FCTR 186
C	J	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	FCTR 188
C			FCTR 190
C	K	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	FCTR 192
C			FCTR 194
C	KCOEF	EARTHQUAKE COEFFICIENT.	FCTR 196
C			FCTR 198
C	MB	CONTROL CODE WHICH INDICATES MODIFIED BISHOP	MAR77EBO -
C		FACTOR OF SAFETY CALCULATION (IF MB=1).	MAR77EBO -
C			MAR77EBO -
C	ND	DO LOOP TERMINATOR.	FCTR 200
C			FCTR 202
C	NDIREC	ARRAY CONTAINING THE NUMBER OF DIRECTION RANGES FOR	FCTR 204
C		EACH ANISOTROPIC SOIL TYPE.	FCTR 206
C			FCTR 208
C	NSAL	NUMBER OF SOIL TYPES HAVING ANISOTROPIC STRENGTH	FCTR 210
C		PROPERTIES.	FCTR 212
C			FCTR 214
C	NSLICE	NUMBER OF SLICES WHICH THE SLIDING MASS IS DIVIDED.	FCTR 216
C			FCTR 218
C	NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	FCTR 220
C			FCTR 222
C	P	ARRAY CONTAINING VALUES OF THE BOUNDARY LOAD APPLIED	FCTR 224
C		TO EACH SLICE.	FCTR 226
C			FCTR 228
C	PHI	ARRAY CONTAINING VALUES OF THE MOHR-COULOMB ANGLE OF	FCTR 230
C		INTERNAL FRICTION FOR EACH SOIL TYPE.	FCTR 232
C			FCTR 234
C	PHIA	ARRAY CONTAINING VALUES OF THE MOHR-COULOMB ANGLE OF	FCTR 236
C		INTERNAL FRICTION FOR EACH SPECIFIED DIRECTION RANGE	FCTR 238
C		OF EACH ANISOTROPIC SOIL TYPE.	FCTR 240
C			FCTR 242
C	RD	FACTOR FOR CONVERSION OF DEGREES TO RADIANS.	FCTR 244
C			FCTR 246
C	SA	SINE OF ANGLE ALPHA.	FCTR 248
C			FCTR 250
C	SB	SINE OF ANGLE BETA.	FCTR 252
C			FCTR 254
C	SD	SINE OF ANGLE DELT.	FCTR 256
C			FCTR 258
C	SIN	STANDARD FUNCTION THAT CALCULATES THE SINE OF AN	FCTR 260
C		ANGLE.	FCTR 262
C			FCTR 264


```

C      SLTP      ARRAY CONTAINING INDICES OF THE SOIL TYPE AT THE BASE OF EACH SLICE. FCTR 266
C      SOILTP     VARIABLE CONTAINING SOIL TYPE AT THE BASE OF A SLICE. FCTR 272
C      SUMB       DENOMINATOR OF FIRST DERIVATIVE OF THE FACTOR OF SAFETY EXPRESSION. FCTR 278
C      SUMT       NUMERATOR OF FIRST DERIVATIVE OF THE FACTOR OF SAFETY EXPRESSION. FCTR 284
C      SURF       ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS DEFINING A TRIAL FAILURE SURFACE. FCTR 290
C      TA         TANGENT OF ANGLE ALPHA. FCTR 296
C      TAN        STANDARD FUNCTION THAT CALCULATES THE SINE OF AN ANGLE. FCTR 300
C      TP         TANGENT OF ANGLE PHI. FCTR 306
C      UALPHA     ARRAY CONTAINING VALUES OF THE HYDROSTATIC FORCE ACTING AT THE BASE OF EACH SLICE. FCTR 310
C      UBETA      ARRAY CONTAINING VALUES OF THE HYDROSTATIC FORCE ACTING AT THE TOP OF EACH SLICE. FCTR 316
C      UKCOEF     VERTICAL EARTHQUAKE COEFFICIENT. FCTR 322
C      WTT        ARRAY CONTAINING VALUES OF THE WEIGHT OF EACH SLICE. FCTR 326
C      ----- FCTR 330
C      ----- FCTR 332
C      COMMON /BLK01/ IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPL0T,IREAD,ISEARC, FCTR 336
C      1 IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL FCTR 338
C      COMMON /BLK02/ BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NEND, FCTR 340
C      1 NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20) FCTR 342
C      COMMON /BLK05/ NSURF,SURF(100,2) FCTR 344
C      COMMON /BLK07/ CSA(10,10),DIREC(10,10),ITPA(10),NDIREC(10),NSAL, FCTR 346
C      1 PHIA(10,10) FCTR 348
C      COMMON /BLK08/ NSLICE,X(300) FCTR 350
C      COMMON /BLK09/ ALPHA(200),BETA(200),DX(200),SLTP(200),UALPHA(200), FCTR 352
C      1 UBETA(200),WTT(200) FCTR 354
C      COMMON /BLK10/ DELT(200),P(200) FCTR 356
C      COMMON /BLK11/ CAUT,KCOEF,UKCOEF FCTR 358
C      COMMON /BLK12/ ANGSI,ANGS2,BPT,EPT,FRTYFU,FS,FSS(12),JJ,NSURFS(12), FCTR 360
C      1 PERPEN,SURFS(100,2,12),TSURF,YBPT,YEPT,YMIN FCTR 362
C      COMMON /BLK15/ M,MB FCTR 364
C      DIMENSION A1(200),A2(200),A3(200) FCTR 366
C      EQUIVALENCE (ALPHA(1),A1(1)), (BETA(1),A2(1)), (DX(1),A3(1)) FCTR 368
C      INTEGER SLTP,SOILTP FCTR 370
C      REAL KCOEF FCTR 372
C      F1=0. FCTR 374
C      ----- FCTR 376
C      CALCULATE REQUIRED ANGLE FUNCTIONS FCTR 378
C      ----- FCTR 380
C      DO 2 I=1,NSLICE FCTR 382
C      SD=SIN(DELT(I)) FCTR 384
C      CD=COS(DELT(I)) FCTR 386
C      SB=SIN(BETA(I)) FCTR 388
C      CB=COS(BETA(I)) FCTR 390
C      SA=SIN(ALPHA(I)) FCTR 392
C      CA=COS(ALPHA(I)) FCTR 394
C      TA=SA/CA FCTR 396

```



```

SOILTP=SLTP(I)
TP=TAN(PHI(SOILTP)*RD)
CSLICE=C(SOILTP)
IF(ISTR.EQ.0)GO TO 3
DO 5 J=1,NSAL
IF(SLTP(I).NE.ITPA(J))GO TO 5
ND=NDIREC(J)
IF(ALPHA(I).GE.DIREC(J,1))GO TO 8
TP=TAN(PHIA(J,1)*RD)
CSLICE=CSA(J,1)
GO TO 3
8 DO 4 K=2,ND
IF((ALPHA(I).LT.DIREC(J,K-1)).OR.(ALPHA(I).GE.DIREC(J,K)))GO TO 4
TP=TAN(PHIA(J,K)*RD)
CSLICE=CSA(J,K)
GO TO 3
4 CONTINUE
5 CONTINUE
-----
CHECK IF THE DENOMINATOR OF THE EXPRESSION FOR
THE EFFECTIVE NORMAL FORCE IS ZERO OR NEGATIVE
-----
3 F=-TA*TP
IF(F.GT.F1)F1=F
-----
CALCULATE A-TERMS REQUIRED FOR FACTOR OF SAFETY CALCULATION
-----
A1(I)=CSLICE*DX(I)/CA**2+TP*(WTT(I)*(1.-KCOEF*TA-UKCOEF)-UALPHA(I)
1 /CA+UBETA(I)*(CB+SB*TA)+P(I)*(CD+SD*TA))
A2(I)=WTT(I)*(TA*KCOEF-UKCOEF*TA)+UBETA(I)*(CB*TA-SB)+P(I)*(CD*TA-
1 SD)
IF(MB.EQ.1) A1(I)=A1(I)*CA
IF(MB.EQ.1) A2(I)=A2(I)*CA
A3(I)=TA*TP
2 CONTINUE
-----
CALCULATE THE FACTOR OF SAFETY
BY THE NEWTON RAPHSON METHOD
-----
FOLD=1.5
DO 10 J=1,10
SUMT=0.
SUMB=0.
DO 6 I=1,NSLICE
DOT=FOLD+A3(I)
SUMT=SUMT+(A1(I)-FOLD*A2(I))/BOT
SUMB=SUMB+(A1(I)+A2(I)*A3(I))/BOT**2
6 CONTINUE
FNEW=FOLD+SUMT/SUMB
-----
CHECK FOR CONVERGENCE
-----
IF(ABS(FNEW-FOLD).LT..005)GO TO 15
10 FOLD=FNEW

```

FCTR 400
FCTR 402
FCTR 404
FCTR 406
FCTR 408
FCTR 410
FCTR 412
FCTR 414 -
FCTR 416
FCTR 418
FCTR 420 -
FCTR 422
FCTR 424
FCTR 426
FCTR 428
FCTR 429 -
FCTR 430
FCTR 432
FCTR 434
FCTR 436
FCTR 438
FCTR 440
FCTR 442
FCTR 444
FCTR 446
FCTR 448
FCTR 450
FCTR 452
FCTR 454
FCTR 456
FCTR 458
FCTR 460
FCTR 462
FCTR 464
FCTR 466
FCTR 467 -
FCTR 468 -
FCTR 469
FCTR 470
FCTR 472
FCTR 474
FCTR 476
FCTR 478
FCTR 480
FCTR 482
FCTR 484
FCTR 486
FCTR 488
FCTR 490
FCTR 492
FCTR 494
FCTR 496
FCTR 498
FCTR 500
FCTR 502
FCTR 504
FCTR 506
FCTR 508
FCTR 510
FCTR 512
FCTR 514
FCTR 516
FCTR 518

C	-----	FCTR 520
C	STATEMENT OF LACK OF CONVERGENCE	FCTR 522
C	-----	FCTR 524
C		FCTR 526
	WRITE(6,103)	FCTR 528
103	FORMAT(//,	FCTR 530
	110X, #FACTOR OF SAFETY CALCULATION HAS GONE THROUGH TEN ITERATIONS#	FCTR 532
	1//)	FCTR 534
	FS=FNEW	FCTR 536
	FNEW= 500.	FCTR 537
	IF(ISEARC.EQ.0) WRITE(6,101)FS	FCTR 538
	IF(ISEARC.EQ.0)GO TO 15	FCTR 539
	WRITE(6,104)NSURF	FCTR 540
104	FORMAT(////,	FCTR 542
	110X, #THE TRIAL FAILURE SURFACE IN QUESTION IS DEFINED#,,	FCTR 544
	110X, #BY THE FOLLOWING#, I3,	FCTR 546
	1 # COORDINATE POINTS#,,//,	FCTR 548
	1 12X, #POINT#, 6X, #X-SURF#, 6X, #Y-SURF#, /	FCTR 550
	1 13X, #NO. #, 8X, #(FT)#, 8X, #(FT)#, /)	FCTR 552
	WRITE(6,105)(I, SURF(I,1), SURF(I,2), I=1, NSURF)	FCTR 554
105	FORMAT(12X, I3, 2X, 2F12.2)	FCTR 556
	WRITE(6,101)FS	FCTR 557
C		FCTR 558
C	-----	FCTR 560
C	PRINT FACTOR OF SAFETY	FCTR 562
C	-----	FCTR 564
C		FCTR 566
	15 FS=FNEW	FCTR 568
	IF(ISEARC.EQ.1)GO TO 7	FCTR 570
	WRITE(6,101)FS	FCTR 572
101	FORMAT(////, 10X, #FACTOR OF SAFETY FOR THE PRECEDING SPECIFIED#,	FCTR 574
	1 # SURFACE #, F7.3)	FCTR 576
	IF(MB.EQ.1) WRITE(6,107)	FCTR 577
107	FORMAT(//, 10X, #WARNING - FACTOR OF SAFETY IS CALCULATED BY THE #,	FCTR 578
	1 #MODIFIED BISHOP#, /, 20X, #METHOD. THIS METHOD IS VALID ONLY #,	FCTR 579
	1 #IF THE FAILURE SURFACE#, /, 20X, #APPROXIMATES A CIRCLE.#)	FCTR 580
	IF(FS.GT.F1)RETURN	FCTR 581
	WRITE(6,106)	FCTR 582
106	FORMAT(//,	FCTR 583
	110X**** THE ABOVE FACTOR OF SAFETY IS MISLEADING ****#)	FCTR 584
	RETURN	FCTR 586
	7 IF(FS.GT.F1)RETURN	FCTR 588
	WRITE(6,102)NSURF	FCTR 590
	WRITE(6,105)(I, SURF(I,1), SURF(I,2), I=1, NSURF)	FCTR 591
102	FORMAT(//,	FCTR 592
	110X#THE FACTOR OF SAFETY FOR THE TRIAL FAILURE SURFACE DEFINED#	FCTR 594
	110X#BY THE COORDINATES LISTED BELOW IS MISLEADING.#///	FCTR 596
	110X#FAILURE SURFACE DEFINED BY#, I3, # COORDINATE POINTS#///	FCTR 598
	112X#POINT#, 6X, #X-SURF#, 6X, #Y-SURF#, /	FCTR 600
	113X#NO. #, 8X, #(FT)#, 8X, #(FT)#, /)	FCTR 602
	WRITE(6,101)FS	FCTR 603
	FS= 500.	FCTR 604
	RETURN	FCTR 605
	END	FCTR 606

SUBROUTINE SCALER		SCAL	2
-----		SCAL	4
-----		SCAL	6
* SUBROUTINE SCALER		SCAL	8
-----		SCAL	10
-----		SCAL	12
-----		SCAL	14
FUNCTIONS -		SCAL	16
		SCAL	18
DETERMINES THE SCALE REQUIRED TO FIT PROBLEM GEOMETRY TO PLOT		SCAL	20
SIZE.		SCAL	22
-----		SCAL	24
-----		SCAL	26
-----		SCAL	28
DEFINITIONS -		SCAL	30
-----		SCAL	32
BND5	AN ARRAY CONTAINING THE COORDINATES OF THE END POINTS	SCAL	34
	DEFINING THE GROUND SURFACE AND SUBSURFACE	SCAL	36
	BOUNDARIES.	SCAL	38
		SCAL	40
I	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	SCAL	42
		SCAL	44
ILIMIT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	SCAL	46
	AFFECTED BY SEARCHING LIMITS ESTABLISHED BY	SCAL	48
	SUBROUTINE LIMITS.	SCAL	50
		SCAL	52
ISURC	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	SCAL	54
	HANDLING THE SURCHARGE LOADS SPECIFIED BY SUBROUTINE	SCAL	56
	LOADS.	SCAL	58
		SCAL	60
IWAT	CONTROL CODE WHICH ACTIVATES PORTIONS OF THE PROGRAM	SCAL	62
	HANDLING THE WATER SURFACE SPECIFIED BY SUBROUTINE	SCAL	64
	WATER.	SCAL	66
		SCAL	68
LIMIT	ARRAY CONTAINING THE COORDINATES OF THE END POINTS	SCAL	70
	DEFINING EACH BOUNDARY USED TO CONFINE THE EXTENT OF	SCAL	72
	SEARCHING BY BLOCK AND RANDOM.	SCAL	74
		SCAL	76
M	VARIABLE USED TO TEMPORARY STORE NPIEZ.	NOU76EB0	-
		NOU76EB0	-
MAXX	MAXIMUM X COORDINATE OF ANY GEOMETRY POINT.	SCAL	78
		SCAL	80
MAXY	MAXIMUM Y COORDINATE OF ANY GEOMETRY POINT.	SCAL	82
		SCAL	84
N	INDEX VARIABLE FOR ARRAY SUBSCRIPTING.	JUL76EB0	-
		JUL76EB0	-
NBND	TOTAL NUMBER OF PROFILE BOUNDARIES.	SCAL	86
		SCAL	88
NLIMIT	NUMBER OF SURFACE GENERATION LIMIT BOUNDARIES.	SCAL	90
		SCAL	92
NPIEZ	NUMBER OF POINTS DEFINING THE WATER SURFACE.	SCAL	94
		SCAL	96
NPZ	NUMBER OF PIEZOMETRIC SURFACES DEFINED.	JUL76EB0	-
		JUL76EB0	-
NSURC	NUMBER OF BOUNDARY LOADS SPECIFIED.	SCAL	98
		SCAL	100
NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	SCAL	102
		SCAL	104
NTOP1	NTOP + 1	SCAL	106
		SCAL	108
SCLE	SCALE USED FOR PLOTTING.	SCAL	110
		SCAL	112
SURC	ARRAY CONTAINING X COORDINATES OF THE END POINTS	SCAL	114
	DEFINING THE EXTENT OF LOADING.	SCAL	116
		SCAL	118


```

C      XPIEZ      ARRAY CONTAINING X COORDINATES OF POINTS DEFINING      SCAL 120
C      WATER SURFACE.                                                    SCAL 122
C                                                                           SCAL 124
C      Y          Y COORDINATE OF A BOUNDARY LOAD END POINT MAGNIFIED      SCAL 126
C      SO SCALE CAN BE ADJUSTED TO FIT BOUNDARY LOADS ON THE SCAL 128
C      PLOT IF NECESSARY.                                                SCAL 130
C                                                                           SCAL 132
C      YSURC      ARRAY CONTAINING THE CALCULATED Y COORDINATES OF THE SCAL 134
C      ENDS OF THE BOUNDARY LOADS ON THE GROUND SURFACE.                SCAL 136
C                                                                           SCAL 138
C      ----- SCAL 140
C      ----- SCAL 142
C      ----- SCAL 144
COMMON /BLK01/IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLOT,IREAD,ISEARC, SCAL 146
1  IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL SCAL 146 -
COMMON /BLK02/BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND, SCAL 150
1  NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20) SCAL 152 -
COMMON /BLK03/UWAT,NPZ,NPIEZ(10),XPIEZ(10,40),YPIEZ(10,40) SCAL 154 -
COMMON /BLK04/DELTA(10),LOAD(10),NSURC,SURC(10,2),YSURC(10,2) SCAL 156
COMMON /BLK05/NSURF,SURF(100,2) SCAL 158
COMMON /BLK06/LIMIT(20,4),NLIMIT,NLMT SCAL 160
COMMON /BLK12/ANGS1,ANGS2,BPT,EPT,FRTYFU,FS,FSS(12),JJ,NSURFS(12), SCAL 162
1  PERPEN,SURFS(100,2,12),TSURF,YBPT,YEPT,YMIN SCAL 164
COMMON /BLK14/SCLE SCAL 166
REAL MAXX,MAXY,LIMIT SCAL 168
C                                                                           SCAL 170
C      ----- SCAL 172
C      DETERMINE SCALE OF PLOTS SCAL 174
C      ----- SCAL 176
C                                                                           SCAL 178
MAXX=BNDS(NTOP,3) SCAL 180
MAXY=BNDS(1,2) SCAL 182
DO 8 I=1,NTOP SCAL 184
IF(BNDS(I,4).GT.MAXY)MAXY=BNDS(I,4) SCAL 186
8 CONTINUE SCAL 188
IF(NTOP.EQ.NBND)GO TO 3 SCAL 190
NTOP1=NTOP+1 SCAL 192
DO 2 I=NTOP1,NBND SCAL 194
IF(BNDS(I,3).GT.MAXX)MAXX=BNDS(I,3) SCAL 196
2 CONTINUE SCAL 198
3 IF(IWAT.EQ.0)GO TO 4 SCAL 199
DO 10 N=1,NPZ SCAL 200
M=NPIEZ(N) SCAL 201
IF(XPIEZ(N,M).GT.MAXX)MAXX=XPIEZ(N,M) SCAL 202
10 CONTINUE SCAL 203
4 IF(ILIMIT.EQ.0)GO TO 5 SCAL 204
DO 6 I=1,NLIMIT SCAL 206
IF(LIMIT(I,3).GT.MAXX)MAXX=LIMIT(I,3) SCAL 208
6 CONTINUE SCAL 210
5 IF(ISURC.EQ.0)GO TO 7 SCAL 212
IF(SURC(NSURC,2).GT.MAXX)MAXX=SURC(NSURC,2) SCAL 214
DO 9 I=1,NSURC SCAL 216
Y=YSURC(I,1)*50./47. SCAL 218
IF(Y.GT.MAXY)MAXY=Y SCAL 220
Y=YSURC(I,2)*50./47. SCAL 222
IF(Y.GT.MAXY)MAXY=Y SCAL 224
9 CONTINUE SCAL 226
7 SCLE=MAXX/8. SCAL 228
MAXX=5./8.*MAXX SCAL 230
C                                                                           SCAL 232
C      ----- SCAL 234
C      CHECK IF VERTICAL SCALE CONTROLS SCAL 236
C      ----- SCAL 238
C                                                                           SCAL 240
IF(MAXY.GT.MAXX)SCLE=MAXY/5. SCAL 242
RETURN SCAL 244
END SCAL 246

```


C	MAXL	MAXIMUM INTENSITY OF THE BOUNDARY LOADS SPECIFIED.	PLOT 130
C			PLOT 132
C	N	VARIABLE USED TO TEMPORARILY STORE THE NUMBER OF	PLOT 134
C		POINTS DEFINING A TRIAL FAILURE SURFACE. ALSO USED	PLOT 136 -
C		AS INDEX VARIABLE FOR ARRAY SUBSCRIPTION.	JUL76EBO
C	NBND	TOTAL NUMBER OF PROFILE BOUNDARIES.	PLOT 138
C			PLOT 140
C	NGRID	NUMBER OF BOXES SPECIFIED.	PLOT 142
C			PLOT 144
C	NLIMIT	NUMBER OF SURFACE GENERATION LIMIT BOUNDARIES.	PLOT 146
C			PLOT 148
C	NN	VARIABLE USED TO TEMPORARY STORE AN ARRAY VARIABLE	PLOT 150
C		FOR A DO STATEMENT.	AUG76EBO -
C			AUG76EBO -
C	NPIEZ	NUMBER OF POINTS DEFINING THE WATER SURFACE.	AUG76EBO -
C			PLOT 152
C	NPZ	NUMBER OF PIEZOMETRIC SURFACES DEFINED.	PLOT 154
C			JUL76EBO -
C	NSURC	NUMBER OF BOUNDARY LOADS SPECIFIED.	JUL76EBO -
C			PLOT 156
C	NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	PLOT 158
C			PLOT 160
C	NSURFS	ARRAY CONTAINING VALUES OF THE NUMBER OF POINTS	PLOT 162
C		DEFINING EACH OF THE TEN MOST CRITICAL TRIAL	PLOT 164
C		SURFACES.	PLOT 166
C			PLOT 168
C	NTOP	NUMBER OF GROUND SURFACE BOUNDARIES.	PLOT 170
C			PLOT 172
C	NTOP1	NTOP + 1	PLOT 174
C			PLOT 176
C	NUMBER	EXTERNAL SUBROUTINE THAT ACTIVATES THE PLOTTING PEN	PLOT 178
C		TO PLOT NUMBERS.	NOV76EBO -
C			NOV76EBO
C	PL	A SCALED BOUNDARY LOAD.	NOV76EBO -
C			PLOT 180
C	PLOT	EXTERNAL SUBROUTINE USED TO MOVE A PLOTTING PEN TO A	PLOT 182
C		NEW POINT.	PLOT 184
C			PLOT 185
C	SCLE	SCALE USED FOR PLOTTING.	PLOT 188
C			PLOT 190
C	SIN	STANDARD FUNCTION THAT CALCULATES THE SINE OF AN	PLOT 192
C		ANGLE.	PLOT 194
C			PLOT 196
C	SURC	ARRAY CONTAINING X COORDINATES OF THE END POINTS	PLOT 198
C		DEFINING THE EXTENT OF LOADING.	PLOT 200
C			PLOT 202
C	SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS	PLOT 204
C		DEFINING A TRIAL FAILURE SURFACE.	PLOT 206
C			PLOT 208
C	SURFS	ARRAY CONTAINING SETS OF COORDINATE POINTS DEFINING	PLOT 210
C		EACH OF THE TEN MOST CRITICAL TRIAL SURFACES.	PLOT 212
C			PLOT 214
C			PLOT 216
C	SYMBOL	EXTERNAL SUBROUTINE THAT ACTIVATES THE PLOTTING PEN	NOV76EBO
C		TO PLOT CHARACTERS OR ON-CENTER SYMBOLS.	NOV76EBO -
C			NOV76EBO
C	W	SCALED WIDTH OF A BOX FOR FOR SLIDING BLOCK SEARCH.	PLOT 218
C			PLOT 220
C	WIDTH	ARRAY CONTAINING THE VALUES OF THE WIDTH OF EACH	PLOT 222
C		BOX.	PLOT 224
C			PLOT 226
C	X	X COORDINATE OF GEOMETRY POINT TO BE PLOTTED.	PLOT 228
C			PLOT 230
C	XL	ARRAY CONTAINING VALUES OF THE X COORDINATE OF THE	PLOT 232
C		LEFT END OF EACH BOX CENTERLINE.	PLOT 234
C			PLOT 236

C	XPIEZ	ARRAY CONTAINING X-COORDINATES OF POINTS DEFINING WATER SURFACE.	PLOT 238
C			PLOT 240
C	XR	ARRAY CONTAINING VALUES OF THE X COORDINATE OF THE RIGHT END OF EACH BOX CENTERLINE.	PLOT 242
C			PLOT 244
C	X1	SCALED X COORDINATE OF A CORNER OF A BOX FOR SLIDING BLOCK SEARCH.	PLOT 246
C			PLOT 248
C	Y	Y COORDINATE OF GEOMETRY POINT TO BE PLOTTED.	PLOT 250
C			PLOT 252
C	YL	ARRAY CONTAINING VALUES OF THE Y COORDINATE OF THE LEFT END OF EACH BOX CENTERLINE.	PLOT 254
C			PLOT 256
C	YPIEZ	ARRAY CONTAINING Y-COORDINATES OF POINTS DEFINING WATER SURFACE.	PLOT 258
C			PLOT 260
C	YR	ARRAY CONTAINING VALUES OF THE Y COORDINATE OF THE RIGHT END OF EACH BOX CENTERLINE.	PLOT 262
C			PLOT 264
C	YSURC	ARRAY CONTAINING THE CALCULATED Y COORDINATES OF THE ENDS OF THE BOUNDARY LOADS ON THE GROUND SURFACE.	PLOT 266
C			PLOT 268
C	Y1	SCALED Y COORDINATE OF A CORNER OF A BOX FOR SLIDING BLOCK SEARCH.	PLOT 270
C			PLOT 272
C			PLOT 274
C			PLOT 276
C			PLOT 278
C			PLOT 280
C			PLOT 282
C			PLOT 284
C			PLOT 286
C			PLOT 288
C			PLOT 290
C			PLOT 292
C			PLOT 294
C	COMMON /BLK01/	IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPL0T,IREAD,ISEARC,	PLOT 296
C	1	IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL	PLOT 298
C	COMMON /BLK02/	BND\$(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND,	PLOT 300
C	1	NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20)	PLOT 302
C	COMMON /BLK03/	UWAT,HPZ,NPIEZ(10),XPIEZ(10,40),YPIEZ(10,40)	PLOT 304
C	COMMON /BLK04/	DELTA(10),LOAD(10),NSURC,SURC(10,2),YSURC(10,2)	PLOT 306
C	COMMON /BLK05/	NSURF,SURF(100,2)	PLOT 308
C	COMMON /BLK06/	LIMIT(20,4),NLIMIT,NLMT	PLOT 310
C	COMMON /BLK12/	ANGS1,ANGS2,BPT,EPT,FRTYFU,FS,FSS(12),JJ,NSURFS(12),	PLOT 312
C	1	PERPEN,SURFS(100,2,12),TSURF,YBPT,YEPT,YMIN	PLOT 314
C	COMMON /BLK13/	NGRID,WIDTH(10),XL(10),XR(10),YL(10),YR(10)	PLOT 316
C	COMMON /BLK14/	SCLE	PLOT 318
C		EQUIVALENCE (DX,X1),(DY,Y1)	PLOT 320
C		REAL LIMIT,LOAD,MAXL	PLOT 322
C			PLOT 324
C			PLOT 326
C	CHECK IF FIRST PLOT		PLOT 328
C			PLOT 330
C			PLOT 332
C	IF(IPL0T.EQ.0)GO TO 2		PLOT 334
C			PLOT 336
C			PLOT 338
C	IF NOT, MAKE PAGE MARK AND POSITION NEW ORIGIN		PLOT 340
C			PLOT 342
C			PLOT 344
C	CALL PLOT(14.,0.,-3)		PLOT 346
C	CALL PLOT(-2.5,0.,3)		PLOT 348
C	CALL PLOT(-2.5,-1.4,2)		PLOT 350
C	2 IPL0T=1		PLOT 352
C			PLOT 354
C			PLOT 356
C	SET UP AXES		PLOT 358
C			PLOT 360
C			PLOT 362
C	CALL AXIS(0.,0.,6HX-AXIS,-6,8.,0.,0.,SCLE,0)		PLOT 364
C	CALL AXIS(0.,0.,6HY-AXIS,6,5.,90.,0.,SCLE,-1)		PLOT 366
C			PLOT 368

C	-----	PLOT	370
C	PLOT PROFILE BOUNDARIES	PLOT	372
C	-----	PLOT	374
C		PLOT	376
	X=BNDS(1,1)/SCLE	PLOT	378
	Y=BNDS(1,2)/SCLE	PLOT	380
	CALL PLOT(X,Y,3)	PLOT	382
	DO 20 I=1,NTOP	PLOT	384
	X=BNDS(I,3)/SCLE	PLOT	386
	Y=BNDS(I,4)/SCLE	PLOT	388
	CALL PLOT(X,Y,2)	PLOT	390
20	CONTINUE	PLOT	392
	IF(NTOP.EQ.NBND)GO TO 21	PLOT	394
	NTOP1=NTOP+1	PLOT	396
	DO 22 I=NTOP1,NBND	PLOT	398
	X=BNDS(I,1)/SCLE	PLOT	400
	Y=BNDS(I,2)/SCLE	PLOT	402
	CALL PLOT(X,Y,3)	PLOT	404
	X=BNDS(I,3)/SCLE	PLOT	406
	Y=BNDS(I,4)/SCLE	PLOT	408
	CALL PLOT(X,Y,2)	PLOT	410
22	CONTINUE	PLOT	412
21	IF(IWAT.EQ.0)GO TO 23	PLOT	414
C	-----	PLOT	416
C	PLOT WATER SURFACE, IF APPLICABLE	PLOT	418
C	-----	PLOT	420
C		PLOT	422
C		PLOT	424
	DO 27 N=1,NPZ	PLOT	425
	X=XPIEZ(N,1)/SCLE	PLOT	426
	Y=YPIEZ(N,1)/SCLE	PLOT	428
	CALL SYMBOL(X,Y,0.07,1HW,0.,1)	PLOT	429
	CALL PLOT(X,Y,3)	PLOT	430
	NN=NPIEZ(N)	PLOT	431
	DO 24 I=2,NN	PLOT	432
	X=XPIEZ(N,1)/SCLE	PLOT	434
	Y=YPIEZ(N,1)/SCLE	PLOT	436
	CALL PLOT(X,Y,2)	PLOT	437
	CALL SYMBOL(X,Y,0.07,1HW,0.,1)	PLOT	438
	CALL PLOT(X,Y,3)	PLOT	439
24	CONTINUE	PLOT	440
27	CONTINUE	PLOT	441
23	IF(ILIMIT.EQ.0)GO TO 25	PLOT	442
C	-----	PLOT	444
C	PLOT SEARCHING BOUNDARIES, IF APPLICABLE	PLOT	446
C	-----	PLOT	448
C		PLOT	450
C		PLOT	452
	DO 26 I=1,NLIMIT	PLOT	454
	X=LIMIT(I,1)/SCLE	PLOT	456
	Y=LIMIT(I,2)/SCLE	PLOT	458
	CALL SYMBOL(X,Y,0.07,1HL,0.,1)	PLOT	459
	CALL PLOT(X,Y,3)	PLOT	460
	X=LIMIT(I,3)/SCLE	PLOT	462
	Y=LIMIT(I,4)/SCLE	PLOT	464
	CALL PLOT(X,Y,2)	PLOT	465
	CALL SYMBOL(X,Y,0.07,1HL,0.,1)	PLOT	466
	CALL PLOT(X,Y,3)	PLOT	467
26	CONTINUE	PLOT	468
25	IF(ISURC.EQ.0)GO TO 3	PLOT	470
C		PLOT	472

C	-----	PLOT 474
C	PLOT SURCHARGE BOUNDARY LOADS, IF APPLICABLE	PLOT 476
C	-----	PLOT 478
C		PLOT 480
	MAXL=LOAD(1)	PLOT 482
	IF(NSURC.EQ.1)GO TO 4	PLOT 484
	DO 7 I=2,NSURC	PLOT 486
	IF(LOAD(I).GT.MAXL)MAXL=LOAD(I)	PLOT 488
7	CONTINUE	PLOT 490
4	DO 8 I=1,NSURC	PLOT 492
	X=SURC(I,1)/SCLE	PLOT 494
	Y=YSURC(I,1)/SCLE	PLOT 496
	CALL SYMBOL(X,Y,0.07,1HP,0.,1)	PLOT 497 -
	CALL PLOT(X,Y,3)	PLOT 498
	PL=.3*LOAD(I)/MAXL	PLOT 500
	DX=PL*SIN(DELTA(I))	PLOT 502
	DY=PL*COS(DELTA(I))	PLOT 504
	X=X-DX	PLOT 506
	Y=Y+DY	PLOT 508
	CALL PLOT(X,Y,2)	PLOT 509
	CALL SYMBOL(X,Y,0.07,1HP,0.,1)	PLOT 510 -
	CALL PLOT(X,Y,3)	PLOT 511 -
	X=SURC(I,2)/SCLE-DX	PLOT 512
	Y=YSURC(I,2)/SCLE+DY	PLOT 514
	CALL PLOT(X,Y,2)	PLOT 515
	CALL SYMBOL(X,Y,0.07,1HP,0.,1)	PLOT 516 -
	CALL PLOT(X,Y,3)	PLOT 517 -
	X=X+DX	PLOT 518
	Y=Y-DY	PLOT 520
	CALL PLOT(X,Y,2)	PLOT 522
	CALL SYMBOL(X,Y,0.07,1HP,0.,1)	PLOT 523 -
8	CONTINUE	PLOT 524
C	-----	PLOT 526
C	PLOT SEARCH BOXES FOR BLOCK, IF APPLICABLE	PLOT 528
C	-----	PLOT 530
C		PLOT 532
C		PLOT 534
	3 IF(IBLK.EQ.0)GO TO 9	PLOT 536
	DO 10 I=1,NGRID	PLOT 538
	W=WIDTH(I)/SCLE	PLOT 540
	X1=XL(I)/SCLE	PLOT 542
	Y1=YL(I)/SCLE+W/2.	PLOT 544
	CALL PLOT(X1,Y1,3)	PLOT 546
	Y=Y1-W	PLOT 548
	CALL PLOT(X1,Y,2)	PLOT 550
	X=XR(I)/SCLE	PLOT 552
	Y=YR(I)/SCLE-W/2.	PLOT 554
	CALL PLOT(X,Y,2)	PLOT 556
	Y=Y+W	PLOT 558
	CALL PLOT(X,Y,2)	PLOT 560
	CALL PLOT(X1,Y1,2)	PLOT 562 -
10	CONTINUE	PLOT 563
	9 IF(ISEARC.EQ.1)RETURN	PLOT 564
	CALL SYMBOL(1.5,5.7,0.1,40HFACTOR OF SAFETY FOR SPECIFIED SURFACE	PLOT 565 -
	1=,0.,40)	PLOT 566 -
	CALL NUMBER(4.95,5.7,0.1,FS,0.,4HF7.3)	PLOT 567 -
C		PLOT 568

C	*****	PLOT 570
	ENTRY PLOTN2	PLOT 572
C	*****	PLOT 574
C		PLOT 576
C	-----	PLOT 578
C	PLOT INDIVIDUAL TRIAL FAILURE SURFACE	PLOT 580
C	-----	PLOT 582
	X=SURF(1,1)/SCLE	PLOT 584
	Y=SURF(1,2)/SCLE	PLOT 586
	CALL PLOT(X,Y,3)	PLOT 588
	DO 5 I=2,NSURF	PLOT 590
	X=SURF(I,1)/SCLE	PLOT 592
	Y=SURF(I,2)/SCLE	PLOT 594
	CALL PLOT(X,Y,2)	PLOT 596
	5 CONTINUE	PLOT 598
	RETURN	PLOT 600
C		PLOT 602
C	*****	PLOT 604
	ENTRY PLOTN3	PLOT 606
C	*****	PLOT 608
C		PLOT 610
C	-----	PLOT 612
C	PLOT THE TEN MOST CRITICAL TRIAL FAILURE SURFACES	PLOT 614
C	-----	PLOT 616
	X=SURFS(1,1,1)/SCLE	PLOT 618
	Y=SURFS(1,2,1)/SCLE	PLOT 620
	CALL SYMBOL(X,Y,0.07,11,0.,-1)	PLOT 622
	N=NSURFS(1)	PLOT 624
	DO 11 I=2,N	PLOT 626
	X=SURFS(I,1,1)/SCLE	PLOT 628
	Y=SURFS(I,2,1)/SCLE	PLOT 630
	CALL SYMBOL(X,Y,0.07,11,0.,-2)	PLOT 632
11	CONTINUE	PLOT 634
	DO 6 K=2,10	PLOT 636
	X=SURFS(1,1,K)/SCLE	PLOT 638
	Y=SURFS(1,2,K)/SCLE	PLOT 640
	CALL PLOT(X,Y,3)	PLOT 642
	N=NSURFS(K)	PLOT 644
	DO 6 I=2,N	PLOT 646
	X=SURFS(I,1,K)/SCLE	PLOT 648
	Y=SURFS(I,2,K)/SCLE	PLOT 650
	CALL PLOT(X,Y,2)	PLOT 652
6	CONTINUE	PLOT 654
	RETURN	PLOT 656
	END	PLOT 658
		PLOT 660
		PLOT 662

C	NLIMIT	NUMBER OF SURFACE GENERATION LIMIT BOUNDARIES.	PLTN 146
C			PLTN 148
C	NN	VARIABLE USED TO TEMPORARY STORE AN ARRAY VARIABLE FOR A DO STATEMENT.	AUG76E80 -
C			AUG76E80 -
C	NPIEZ	NUMBER OF POINTS DEFINING THE WATER SURFACE.	AUG76E80 -
C			PLTN 150
C	NPZ	NUMBER OF PIEZOMETRIC SURFACES DEFINED.	PLTN 152
C			JUL76E80 -
C	NSURC	NUMBER OF BOUNDARY LOADS SPECIFIED.	JUL76E80 -
C			PLTN 154
C	NSURF	NUMBER OF POINTS DEFINING A TRIAL FAILURE SURFACE.	PLTN 156
C			PLTN 158
C	NSURFS	ARRAY CONTAINING VALUES OF THE NUMBER OF POINTS DEFINING EACH OF THE TEN MOST CRITICAL TRIAL SURFACES.	PLTN 160
C			PLTN 162
C			PLTN 164
C			PLTN 166
C	PLT	ARRAY CONTAINING THE CHARACTER PLOT MATRIX.	PLTN 168
C			PLTN 170
C	POSTN	SUBROUTINE WHICH DETERMINES THE POSITION OF A SCALED COORDINATE POINT WITHIN THE CHARACTER PLOT ARRAY.	PLTN 172
C			PLTN 174
C			PLTN 176
C	SCL	ARRAY WHICH CONTAINS THE TICK MARK LABELS.	PLTN 178
C			PLTN 180
C	SCLE	SCALE USED FOR PLOTTING.	PLTN 182
C			PLTN 184
C	SURC	ARRAY CONTAINING X COORDINATES OF THE END POINTS DEFINING THE EXTENT OF LOADING.	PLTN 186
C			PLTN 188
C			PLTN 190
C	SURF	ARRAY CONTAINING THE X AND Y COORDINATES OF POINTS DEFINING A TRIAL FAILURE SURFACE.	PLTN 192
C			PLTN 194
C			PLTN 196
C			PLTN 198
C	SURFS	ARRAY CONTAINING SETS OF COORDINATE POINTS DEFINING EACH OF THE TEN MOST CRITICAL TRIAL SURFACES.	PLTN 200
C			PLTN 202
C			PLTN 204
C	SYMB	ARRAY WHICH CONTAINS THE CHARACTERS THAT ARE USED FOR PLOTTING.	PLTN 206
C			PLTN 208
C			PLTN 210
C	TOL	TOLERANCE CONSTANT TO ACCOUNT FOR MACHINE ROUNDING.	PLTN 212
C			PLTN 214
C	XPIEZ	ARRAY CONTAINING X-COORDINATES OF POINTS DEFINING WATER SURFACE.	PLTN 216
C			PLTN 218
C			PLTN 220
C	YPIEZ	ARRAY CONTAINING Y-COORDINATES OF POINTS DEFINING WATER SURFACE.	PLTN 222
C			PLTN 224
C			PLTN 226
C	YSURC	ARRAY CONTAINING THE CALCULATED Y COORDINATES OF THE ENDS OF THE BOUNDARY LOADS ON THE GROUND SURFACE.	PLTN 228
C			PLTN 230
C			PLTN 232
C			PLTN 234
C			PLTN 236
C			PLTN 238
C	COMMON /BLK01/	IANGL,IBLK,IEXIT,ICIRC,ILIMIT,IPLT,IREAD,ISEARC,	PLTN 240
C	1	IBLK2,ISOIL,ISTR,ISURC,ISURF,IWAT,RD,TOL	PLTN 242 -
C	COMMON /BLK02/	BNDS(100,4),C(20),GAMMA(20),GSAT(20),ITP(100),NBND,	PLTN 244
C	1	NSOIL,NTOP,PHI(20),RU(20),CU(20),NP(20)	PLTN 246 -
C	COMMON /BLK03/	UWAT,NPZ,NPIEZ(10),XPIEZ(10,40),YPIEZ(10,40)	PLTN 248 -
C	COMMON /BLK04/	DELTA(10),LOAD(10),NSURC,SURC(10,2),YSURC(10,2)	PLTN 250
C	COMMON /BLK05/	NSURF,SURF(100,2)	PLTN 252
C	COMMON /BLK06/	LIMIT(20,4),NLIMIT,NLMT	PLTN 254
C	COMMON /BLK12/	ANGS1,ANGS2,BPT,EPT,FRTYFU,FS,FSS(12),JJ,NSURFS(12),	PLTN 256
C	1	PERPEN,SURFS(100,2,12),TSURF,YBPT,YEPT,YMIN	PLTN 258
C	COMMON /BLK14/	SCLE	PLTN 260
C		DIMENSION PLT(49,51),SYMB(19),AXIS(9),SCL(9)	PLTN 262
C		REAL LIMIT	PLTN 264
C		DATA SYMB/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H ,1H+,1H-,1H*,	PLTN 266
C	1	1HW,1HL,1HS,1H.,1H//	PLTN 268
C		DATA AXIS/1HX,1H ,1HA,1HX,1HI,1HS,1H ,1HF,1HT/	PLTN 270

C		PLTN 272
C		PLTN 274
C	-----	PLTN 276
C	SET UP HORIZONTAL AXIS	PLTN 278
C	-----	PLTN 280
	DO 2 I=2,48	PLTN 282
	PLT(I,1)=SYMB(13)	PLTN 284
	2 CONTINUE	PLTN 286
	DO 3 I=1,49,6	PLTN 288
	PLT(I,1)=SYMB(12)	PLTN 290
	3 CONTINUE	PLTN 292
C		PLTN 294
C	-----	PLTN 296
C	SET UP VERTICAL AXIS	PLTN 298
C	-----	PLTN 300
	DO 4 J=2,50	PLTN 302
	PLT(1,J)=SYMB(13)	PLTN 304
	4 CONTINUE	PLTN 306
	DO 5 J=1,51,10	PLTN 308
	PLT(1,J)=SYMB(12)	PLTN 310
	5 CONTINUE	PLTN 312
C		PLTN 314
C	-----	PLTN 316
C	INITIATE REMAINDER OF GRAPH WITH BLANKS	PLTN 318
C	-----	PLTN 320
	DO 6 I=2,49	PLTN 322
	DO 6 J=2,51	PLTN 324
	PLT(I,J)=SYMB(11)	PLTN 326
	6 CONTINUE	PLTN 328
C		PLTN 330
C	-----	PLTN 332
C	CALCULATE TICK MARK LABLES	PLTN 334
C	-----	PLTN 336
	SCL(1)=TOL	PLTN 338
	DO 18 I=2,9	PLTN 340
	SCL(I)=SCL(I-1)+SCLE	PLTN 342
	18 CONTINUE	PLTN 344
	RETURN	PLTN 346
C		PLTN 348
C	*****	PLTN 350
	ENTRY PLT2	PLTN 352
C	*****	PLTN 354
C	-----	PLTN 356
C	POSITION POINTS DEFINING A GENERATED TRIAL FAILURE SURFACE	PLTN 358
C	-----	PLTN 360
	DO 11 I=1,NSURF	PLTN 362
	CALL POSTN(SURF(I,1),SURF(I,2),IX,IY)	PLTN 364
	PLT(IX,IY)=SYMB(18)	PLTN 366
	11 CONTINUE	PLTN 368
	RETURN	PLTN 370
C		PLTN 372
C	*****	PLTN 374
	ENTRY PLT3	PLTN 376
C	*****	PLTN 378
C	-----	PLTN 380
C	POSITION POINTS DEFINING THE TEN MOST CRITICAL	PLTN 382
C	TRIAL FAILURE SURFACES, IF APPLICABLE	PLTN 384
C	-----	PLTN 386
	DO 14 K=1,10	PLTN 388
C		PLTN 390
		PLTN 392
		PLTN 394
		PLTN 396
		PLTN 398
		PLTN 400
		PLTN 402
		PLTN 404

C	-----	PLTN 406
C	POSITION IN ORDER OF LEAST CRITICAL FIRST	PLTN 408
C	-----	PLTN 410
C		PLTN 412
	KK=11-K	PLTN 414
	N=NSURFS(KK)	PLTN 416
	DO 14 I=1,N	PLTN 416
	CALL POSTN(SURFS(I,1,KK),SURFS(I,2,KK),IX,IY)	PLTN 420
	PLT(IX,IY)=SYMB(KK)	PLTN 422
	14 CONTINUE	PLTN 424
	RETURN	PLTN 426
C		PLTN 428
C	*****	PLTN 430
	ENTRY PLT4	PLTN 432
C	*****	PLTN 434
C		PLTN 436
	IF(IWAT.EQ.0)GO TO 9	PLTN 438
C		PLTN 440
C	-----	PLTN 442
C	POSITION POINTS DEFINING GROUNDWATER TABLE, IF APPLICABLE	PLTN 444
C	-----	PLTN 446
	DO 22 N=1,NPZ	PLTN 447
	NN=NPIEZ(N)	PLTN 448
	DO 7 I=1,NN	PLTN 449
	CALL POSTN(XPIEZ(N,I),YPIEZ(N,I),IX,IY)	PLTN 450
	PLT(IX,IY)=SYMB(15)	PLTN 452
	7 CONTINUE	PLTN 454
	22 CONTINUE	PLTN 456
	9 IF(ILIMIT.EQ.0)GO TO 12	PLTN 457
C		PLTN 458
C	-----	PLTN 460
C	POSITION BOUNDARY END POINTS DEFINING	PLTN 462
C	SEARCHING LIMITS, IF APPLICABLE	PLTN 464
C	-----	PLTN 466
	DO 13 I=1,NLIMIT	PLTN 468
	DO 13 J=2,4,2	PLTN 470
	CALL POSTN(LIMIT(I,J-1),LIMIT(I,J),IX,IY)	PLTN 472
	PLT(IX,IY)=SYMB(16)	PLTN 474
	13 CONTINUE	PLTN 476
	12 IF(ISEARC.NE.0)GO TO 21	PLTN 478
C		PLTN 480
C	-----	PLTN 482
C	POSITION POINTS DEFINING A SPECIFIED TRIAL FAILURE SURFACE	PLTN 484
C	-----	PLTN 486
	DO 20 I=1,NSURF	PLTN 488
	CALL POSTN(SURF(I,1),SURF(I,2),IX,IY)	PLTN 490
	PLT(IX,IY)=SYMB(17)	PLTN 492
	20 CONTINUE	PLTN 494
	21 IF(ISURC.EQ.0)GO TO 10	PLTN 496
C		PLTN 498
C	-----	PLTN 500
C	POSITION POINTS DEFINING EXTENT OF SURCHARGE BOUNDARY LOADS	PLTN 502
C	-----	PLTN 504
	CALL POSTN(SURC(1,1),YSURC(1,1),IX,IY)	PLTN 506
	PLT(IX,IY+2)=SYMB(1)	PLTN 508
	PLT(IX,IY+1)=SYMB(19)	PLTN 510
	IXX=IX	PLTN 512
	IYY=IY	PLTN 514
	CALL POSTN(SURC(1,2),YSURC(1,2),IX,IY)	PLTN 516
	IF(IX.GT.IXX)GO TO 15	PLTN 518
	PLT(IXX,IYY+1)=SYMB(1)	PLTN 520
	PLT(IXX,IYY+2)=SYMB(19)	PLTN 522
	PLT(IXX,IYY+3)=SYMB(1)	PLTN 524
	GO TO 17	PLTN 526
		PLTN 528
		PLTN 530
		PLTN 532
		PLTN 534

15	PLT(IX,IY+1)=SYMB(1)	PLTN	536
	PLT(IX,IY+2)=SYMB(19)	PLTN	538
	IXX=IX	PLTN	540
	IYY=IY	PLTN	542
17	IF(NSURC.EQ.1)GO TO 10	PLTN	544
	DO 23 I=2,NSURC	PLTN	546
	CALL POSTN(SURC(I,1),YSURC(I,1),IX,IY)	PLTN	548
	IF(IX.GT.IXX)GO TO 24	PLTN	550
	PLT(IXX,IYY+1)=SYMB(I-1)	PLTN	552
	PLT(IXX,IYY+2)=SYMB(19)	PLTN	554
	PLT(IXX,IYY+3)=SYMB(I)	PLTN	556
	GO TO 25	PLTN	558
24	PLT(IX,IY+2)=SYMB(I)	PLTN	560
	PLT(IX,IY+1)=SYMB(19)	PLTN	562
	IXX=IX	PLTN	564
	IYY=IY	PLTN	566
25	CALL POSTN(SURC(I,2),YSURC(I,2),IX,IY)	PLTN	568
	IF(IX.GT.IXX)GO TO 26	PLTN	570
	PLT(IXX,IYY+1)=SYMB(I)	PLTN	572
	PLT(IXX,IYY+2)=SYMB(19)	PLTN	574
	PLT(IXX,IYY+3)=SYMB(I)	PLTN	576
	GO TO 23	PLTN	578
26	PLT(IX,IY+1)=SYMB(I)	PLTN	580
	PLT(IX,IY+2)=SYMB(19)	PLTN	582
	IXX=IX	PLTN	584
	IYY=IY	PLTN	586
23	CONTINUE	PLTN	588
C		PLTN	590
C	-----	PLTN	592
C	POSITION END POINTS DEFINING PROFILE BOUNDARIES	PLTN	594
C	-----	PLTN	596
C		PLTN	598
	10 DO 8 I=1,NBND	PLTN	600
	DO 8 J=2,4,2	PLTN	602
	CALL POSTN(BNDS(I,J-1),BNDS(I,J),IX,IY)	PLTN	604
	PLT(IX,IY)=SYMB(14)	PLTN	606
	8 CONTINUE	PLTN	608
C		PLTN	610
C	-----	PLTN	612
C	PRINT CHARACTER GRAPH	PLTN	614
C	-----	PLTN	616
C		PLTN	618
	WRITE(6,101)(SCL(I),I=1,6)	PLTN	620
101	FORMAT(1H1,21X,*,Y*,12X,*,A X I S*,12X,*,F T*,///,	PLTN	622
	1 15X,6F10.2,/,)	PLTN	624
	II=0	PLTN	626
	DO 16 I=1,49,6	PLTN	628
	II=II+1	PLTN	630
	WRITE(6,102)AXIS(II),SCL(II),(PLT(I,J),J=1,51)	PLTN	632
102	FORMAT(10X,A1,F10.2,1X,51A1)	PLTN	634
	IF(I.EQ.49)GO TO 19	PLTN	636
	DO 16 JJ=1,5	PLTN	638
	IJ=I+JJ	PLTN	640
	WRITE(6,103)(PLT(IJ,J),J=1,51)	PLTN	642
103	FORMAT(22X,51A1)	PLTN	644
16	CONTINUE	PLTN	646
19	CONTINUE	PLTN	648
	RETURN	PLTN	650
	END	PLTN	652

C C C C C C C C C C C C C C C C C C C C C	SUBROUTINE POSTN(XX,YY,IX,IY)	PSTN	2
	-----	PSTN	4
	-----	PSTN	6
	SUBROUTINE POSTN	PSTN	8
	-----	PSTN	10
	-----	PSTN	12
		PSTN	14
	FUNCTION -	PSTN	16
		PSTN	18
	DETERMINES THE POSITION OF A SCALED COORDINATE POINT WITHIN THE	PSTN	20
	CHARACTER PLOT ARRAY.	PSTN	22
		PSTN	24
	-----	PSTN	26
	DEFINITIONS -	PSTN	28
		PSTN	30
		PSTN	32
	FLOAT STANDARD FUNCTION THAT CONVERTS AN INTEGER NUMBER TO	PSTN	34
	A REAL NUMBER.	PSTN	36
		PSTN	38
	IFIX STANDARD FUNCTION THAT CONVERTS A REAL NUMBER TO A	PSTN	40
	INTEGER NUMBER.	PSTN	42
		PSTN	44
	IX HORIZONTAL POSITION OF POINT WITHIN PLOT ARRAY.	PSTN	46
		PSTN	48
	IY VERTICAL POSITION OF POINT WITHIN PLOT ARRAY.	PSTN	50
		PSTN	52
	SCLE SCALE USED FOR PLOTTING.	PSTN	54
		PSTN	56
		PSTN	58
	X X COORDINATE OF SCALED POINT.	PSTN	60
		PSTN	62
	XX X COORDINATE OF POINT TO BE SCALED.	PSTN	64
		PSTN	66
	Y Y COORDINATE OF SCALED POINT.	PSTN	68
		PSTN	70
	YY Y COORDINATE OF POINT TO BE SCALED.	PSTN	72
		PSTN	74
	-----	PSTN	76
	-----	PSTN	78
		PSTN	80
	COMMON /BLK14/SCLE	PSTN	82
	X=XX*6./SCLE+1.	PSTN	84
	Y=YY*10./SCLE+1.	PSTN	86
	IX=IFIX(X)	PSTN	88
	IY=IFIX(Y)	PSTN	90
	IF((X-FLOAT(IX)).GE..5)IX=IX+1	PSTN	92
	IF((Y-FLOAT(IY)).GE..5)IY=IY+1	PSTN	94
	IF(IX.LT.0)IX=0	PSTN	96
	IF(IY.LT.0)IY=0	PSTN	98
	RETURN	PSTN	100
	END	PSTN	102

Cards to Interchange/Add in STABL2
to Use Program on an IBM-Computer

DIMENSION KEYW(15),ERROR(5),WAR(100)	STBL 250	
REAL*8 MKEYW,KEYW	STBL 251	
TOL=.001	STBL 258	STABL2
CALL PLOTS(WAR,400)	STBL 292	Main
READ(5,100,END=19)MKEYW	STBL 310	
IF(I PROF.EQ.1)GO TO 23	STBL 336	
		Subr.
5 READ(5,101,END=23)(M(I),I=1,80)	READ 180	
GO TO 24	READ 194	READER
DIMENSION TITLE(10),DESCR(10), ERROR(11)	PROF 344	
113 FORMAT(20A4)	PROF 440	PROFIL
112 FORMAT(//,10X,PROBLEM DESCRIPTION #,10A4,/32X,10A4,//)	PROF 443	
ENTRY INTSC2(X1,Y1,X2,Y2,X3,Y3,X4,Y4,X,Y,INTS)	ISCT 228	INTSCT
CALL NUMBER (1.5,6.0,0.1,FLOAT(TAL),0.,-1)	RAND1859	RANDOM
CALL NUMBER (3.8,5.7,0.1,FSS(1),0.,3)	RAND1868	
CALL AXIS(0.,0.,6HX-AXIS,-6.8,0.,0.,SCALE,1.,1,1,0.,125)	PLOT 364	PLOTIN
CALL AXIS(0.,0.,6HY-AXIS,6.5,1.5708,0.,SCALE,1.,1,1,0.,125)	PLOT 366	
CALL NUMBER(4.95,5.7,0.1,FS,0.,3)	PLOT 567	
FUNCTION RANF(X)	RANF 2	
INTEGER XN,SEED	RANF 4	Random
DATA XN,SEED/16384,27487/	RANF 6	Function
SEED=MOD(SEED*78125,XN)	RANF 8	
RANF=FLOAT(SEED)/FLOAT(XN)	RANF 10	
RETURN	RANF 12	
END	RANF 14	

Further: Omit STBL 2 and STBL 324 in STABL2 Main.

Note: Some of the above changes reflect adjustment to use the Calcomp plotter on local system (ISHC-IBM360). They may have to be adjusted different for other systems.

APPENDIX F

STABL ON THE IBM COMPUTER

F.1 INTRODUCTION

The STABL program is developed for a CDC 6500 computer at Purdue University. Some changes were necessary to transfer the program to an IBM 360 computer at the Indiana State Highway Commission's Computer Center in Indianapolis.

A search of the STABL program for the errors that caused initial failures on the IBM computer became a part of this research.

A listing of the changes necessary to transfer the STABL program from the CDC 6500 computer to an IBM 360 (or 370) computer is found in Table F. The changes associated with the plotting directives depend on local library plotting routines, and may vary from one computer to another.

F.2 THE RANDOM FUNCTION RANF(X)

The CDC computer has a standard random function $\text{Ranf}(x)$, which in a controlled manner picks a pseudo-random number between 0 and 1. This function is used in STABL to generate failure surfaces randomly within specified limits.

$\text{Ranf}(x)$ is not available from the IBM computer library, so a function similar to the CDC random function was added to the IBM version of STABL. Although the two functions are not exactly the same, the results are usually close.

During the error trace of STABL on the IBM computer, the local $\text{Ranf}(x)$ function in STABL was used on both computers, to make exact comparison possible. It was observed then that the CIRCLE and BLOCK procedures would give exactly the same results on the two computers, but not the RANDOM procedure.

The case was further investigated, and the reason for the difference is as follows. The irregular surface generation routine (RANDOM) uses an expression for generation of line segments that includes the term

$$R^{(1 + R)}$$

where $R = \text{Ranf}(0)$. On the IBM computer, this expression is calculated as

$$R_i^{(1 + R_{i+1})}$$

where R_i is a number in the random sequence, and R_{i+1} is the next number in the sequence. On the CDC computer the same expression is treated as

$$R_i^{(1 + R_i)}$$

so the same random number is used in both places.

F.3 PRECISION

Some of the troubles encountered with STABL on the IBM computer were caused by the difference in precision between the CDC computer and the IBM computer.

F.3.1 Machine Rounding Errors

STABL has a variable, TOL, that is used to account for machine

roundings. This tolerance was originally defined as 10^{-6} , which worked fine on the CDC computer that handles approximately 14 significant digits with single precision and 27 with double. However, the IBM computer is only accurate to 7 digits with single precision and 16 with double precision.

One way to overcome the problem with machine rounding errors is to specify all variables with double precision on the IBM computer. However, this increases the amount of computer memory required to execute the program.

Another way is to increase the tolerance, TOL. It was felt that this could be done without lowering the accuracy of the results. This proved to be correct for the cases investigated during the error trace.

The tolerance was first increased to 10^{-4} , and later to 10^{-3} , which should be sufficient for most problems analysed. However, if the geometry is defined with coordinates exceeding 999.999, the tolerance may have to be increased to 10^{-2} to avoid machine rounding errors.

F.3.2 Use of TOL

Before the tolerance, TOL, was changed, its place and purpose in the program was thoroughly investigated.

First it is used to check order and consistency of data, e.g., if the geometry data are all defined within the first quadrant of the coordinate system (value $> -TOL$). It was particularly here that the errors occurred on the IBM, because the machine roundings were larger than the tolerance.

In subroutine SLICES, TOL is used to eliminate duplicate divisions of the sliding mass into slices. If the difference between two adjacent

division points is less than TOL, only one is used.

In subroutine INTSCT, TOL is used to check for vertical line segments, parallel lines, and order of points.

In the circular surface generator, TOL defines the "straight line" solution in terms of minimum change in angle between adjacent line segments. A TOL of 10^{-3} corresponds to an angle of 0.06 degrees.

Finally, the tolerance is used to define the (0,0) point {as (TOL, TOL)} on the print character plot.

None of these uses suffers by increasing the tolerance, as was done.

.

Table F
Changes Necessary to Transfer STABL from the CDC Computer to an IBM Computer

Change	Card no.	CDC - version	IBM - version	Explanation
1	STBL 2	PROGRAM STABL(INPUT,OUTPUT,TAPES=INPUT,TAPES=OUTPUT,PLOT)	Omit	Not used on the IBM (automatically defined).
2	STBL 230	OPTIONAL KERN(15),ERRORS(5),VAR(100)	DIMENSION KERN(15),ERRORS(5),VAR(100)	VAR specifies a working area needed for plotting on the IBM.
3	STBL 231	-	REAL*8 KERN,KERN	Specifies KERN and KERN as 8-byte words.
4	STBL 238	TOL=.0001	TOL=.001	TOL accounts for machine roundoffs. Due to less precision on the IBM, it has to be increased.
5	STBL 292	CALL PLOTS	CALL PLOTS(VAR,400)	Related to plotting routine.
6	STBL 310	READ(5,100)KERN	READ(5,100,END=19)KERN	EOF = End of File, is a specific CDC feature that does not exist on the IBM.
7	STBL 324	IF(EOF,5,19,50)	Omit	
8	STBL 336	50 IF(EOF,5,19,1)GO TO 23	IF(1PWR,5,1)GO TO 23	
9	READ 180	5 READ(5,101) (H(1),1=1,80)	5 READ(5,101,END=23) (H(1),1=1,80)	
10	READ 194	IF(EOF,5,23,24)	GO TO 24	
11	PROF 344	DIMENSION TITLE(4),DESCR(4),ERRORS(11)	DIMENSION TITLE(10),DESCR(10),FRONT(11)	Related to plotting. The CDC Calcomp plotter requires words defined with A10 format, whereas the IBM only accepts A4 format.
12	PROF 447	113 FORMAT(A10)	113 FORMAT(20A4)	
13	PROF 443	112 FORMAT(//,10X,'PROBLEM DESCRIPTION',A410,/32X,4A10,//)	112 FORMAT(//,10X,'PROBLEM DESCRIPTION',A10A4,/32X,10A4,//)	Dummy variables has to be repeated at Entry points on the IBM. Differences in the library subprogram NAMES used for plotting.
14	ISCT 228	ENTRY INTSC	ENTRY INTSC(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12,X13,X14,X15,X16,X17,X18,X19,X20,X21,X22,X23,X24,X25,X26,X27,X28,X29,X30,X31,X32,X33,X34,X35,X36,X37,X38,X39,X40,X41,X42,X43,X44,X45,X46,X47,X48,X49,X50,X51,X52,X53,X54,X55,X56,X57,X58,X59,X60,X61,X62,X63,X64,X65,X66,X67,X68,X69,X70,X71,X72,X73,X74,X75,X76,X77,X78,X79,X80,X81,X82,X83,X84,X85,X86,X87,X88,X89,X90,X91,X92,X93,X94,X95,X96,X97,X98,X99,X100,X101,X102,X103,X104,X105,X106,X107,X108,X109,X110,X111,X112,X113,X114,X115,X116,X117,X118,X119,X120,X121,X122,X123,X124,X125,X126,X127,X128,X129,X130,X131,X132,X133,X134,X135,X136,X137,X138,X139,X140,X141,X142,X143,X144,X145,X146,X147,X148,X149,X150,X151,X152,X153,X154,X155,X156,X157,X158,X159,X160,X161,X162,X163,X164,X165,X166,X167,X168,X169,X170,X171,X172,X173,X174,X175,X176,X177,X178,X179,X180,X181,X182,X183,X184,X185,X186,X187,X188,X189,X190,X191,X192,X193,X194,X195,X196,X197,X198,X199,X200,X201,X202,X203,X204,X205,X206,X207,X208,X209,X210,X211,X212,X213,X214,X215,X216,X217,X218,X219,X220,X221,X222,X223,X224,X225,X226,X227,X228,X229,X230,X231,X232,X233,X234,X235,X236,X237,X238,X239,X240,X241,X242,X243,X244,X245,X246,X247,X248,X249,X250,X251,X252,X253,X254,X255,X256,X257,X258,X259,X260,X261,X262,X263,X264,X265,X266,X267,X268,X269,X270,X271,X272,X273,X274,X275,X276,X277,X278,X279,X280,X281,X282,X283,X284,X285,X286,X287,X288,X289,X290,X291,X292,X293,X294,X295,X296,X297,X298,X299,X300,X301,X302,X303,X304,X305,X306,X307,X308,X309,X310,X311,X312,X313,X314,X315,X316,X317,X318,X319,X320,X321,X322,X323,X324,X325,X326,X327,X328,X329,X330,X331,X332,X333,X334,X335,X336,X337,X338,X339,X340,X341,X342,X343,X344,X345,X346,X347,X348,X349,X350,X351,X352,X353,X354,X355,X356,X357,X358,X359,X360,X361,X362,X363,X364,X365,X366,X367,X368,X369,X370,X371,X372,X373,X374,X375,X376,X377,X378,X379,X380,X381,X382,X383,X384,X385,X386,X387,X388,X389,X390,X391,X392,X393,X394,X395,X396,X397,X398,X399,X400,X401,X402,X403,X404,X405,X406,X407,X408,X409,X410,X411,X412,X413,X414,X415,X416,X417,X418,X419,X420,X421,X422,X423,X424,X425,X426,X427,X428,X429,X430,X431,X432,X433,X434,X435,X436,X437,X438,X439,X440,X441,X442,X443,X444,X445,X446,X447,X448,X449,X450,X451,X452,X453,X454,X455,X456,X457,X458,X459,X460,X461,X462,X463,X464,X465,X466,X467,X468,X469,X470,X471,X472,X473,X474,X475,X476,X477,X478,X479,X480,X481,X482,X483,X484,X485,X486,X487,X488,X489,X490,X491,X492,X493,X494,X495,X496,X497,X498,X499,X500,X501,X502,X503,X504,X505,X506,X507,X508,X509,X510,X511,X512,X513,X514,X515,X516,X517,X518,X519,X520,X521,X522,X523,X524,X525,X526,X527,X528,X529,X530,X531,X532,X533,X534,X535,X536,X537,X538,X539,X540,X541,X542,X543,X544,X545,X546,X547,X548,X549,X550,X551,X552,X553,X554,X555,X556,X557,X558,X559,X560,X561,X562,X563,X564,X565,X566,X567,X568,X569,X570,X571,X572,X573,X574,X575,X576,X577,X578,X579,X580,X581,X582,X583,X584,X585,X586,X587,X588,X589,X590,X591,X592,X593,X594,X595,X596,X597,X598,X599,X600,X601,X602,X603,X604,X605,X606,X607,X608,X609,X610,X611,X612,X613,X614,X615,X616,X617,X618,X619,X620,X621,X622,X623,X624,X625,X626,X627,X628,X629,X630,X631,X632,X633,X634,X635,X636,X637,X638,X639,X640,X641,X642,X643,X644,X645,X646,X647,X648,X649,X650,X651,X652,X653,X654,X655,X656,X657,X658,X659,X660,X661,X662,X663,X664,X665,X666,X667,X668,X669,X670,X671,X672,X673,X674,X675,X676,X677,X678,X679,X680,X681,X682,X683,X684,X685,X686,X687,X688,X689,X690,X691,X692,X693,X694,X695,X696,X697,X698,X699,X700,X701,X702,X703,X704,X705,X706,X707,X708,X709,X710,X711,X712,X713,X714,X715,X716,X717,X718,X719,X720,X721,X722,X723,X724,X725,X726,X727,X728,X729,X730,X731,X732,X733,X734,X735,X736,X737,X738,X739,X740,X741,X742,X743,X744,X745,X746,X747,X748,X749,X750,X751,X752,X753,X754,X755,X756,X757,X758,X759,X760,X761,X762,X763,X764,X765,X766,X767,X768,X769,X770,X771,X772,X773,X774,X775,X776,X777,X778,X779,X780,X781,X782,X783,X784,X785,X786,X787,X788,X789,X790,X791,X792,X793,X794,X795,X796,X797,X798,X799,X800,X801,X802,X803,X804,X805,X806,X807,X808,X809,X810,X811,X812,X813,X814,X815,X816,X817,X818,X819,X820,X821,X822,X823,X824,X825,X826,X827,X828,X829,X830,X831,X832,X833,X834,X835,X836,X837,X838,X839,X840,X841,X842,X843,X844,X845,X846,X847,X848,X849,X850,X851,X852,X853,X854,X855,X856,X857,X858,X859,X860,X861,X862,X863,X864,X865,X866,X867,X868,X869,X870,X871,X872,X873,X874,X875,X876,X877,X878,X879,X880,X881,X882,X883,X884,X885,X886,X887,X888,X889,X890,X891,X892,X893,X894,X895,X896,X897,X898,X899,X900,X901,X902,X903,X904,X905,X906,X907,X908,X909,X910,X911,X912,X913,X914,X915,X916,X917,X918,X919,X920,X921,X922,X923,X924,X925,X926,X927,X928,X929,X930,X931,X932,X933,X934,X935,X936,X937,X938,X939,X940,X941,X942,X943,X944,X945,X946,X947,X948,X949,X950,X951,X952,X953,X954,X955,X956,X957,X958,X959,X960,X961,X962,X963,X964,X965,X966,X967,X968,X969,X970,X971,X972,X973,X974,X975,X976,X977,X978,X979,X980,X981,X982,X983,X984,X985,X986,X987,X988,X989,X990,X991,X992,X993,X994,X995,X996,X997,X998,X999,X1000,X1001,X1002,X1003,X1004,X1005,X1006,X1007,X1008,X1009,X1010,X1011,X1012,X1013,X1014,X1015,X1016,X1017,X1018,X1019,X1020,X1021,X1022,X1023,X1024,X1025,X1026,X1027,X1028,X1029,X1030,X1031,X1032,X1033,X1034,X1035,X1036,X1037,X1038,X1039,X1040,X1041,X1042,X1043,X1044,X1045,X1046,X1047,X1048,X1049,X1050,X1051,X1052,X1053,X1054,X1055,X1056,X1057,X1058,X1059,X1060,X1061,X1062,X1063,X1064,X1065,X1066,X1067,X1068,X1069,X1070,X1071,X1072,X1073,X1074,X1075,X1076,X1077,X1078,X1079,X1080,X1081,X1082,X1083,X1084,X1085,X1086,X1087,X1088,X1089,X1090,X1091,X1092,X1093,X1094,X1095,X1096,X1097,X1098,X1099,X1100,X1101,X1102,X1103,X1104,X1105,X1106,X1107,X1108,X1109,X1110,X1111,X1112,X1113,X1114,X1115,X1116,X1117,X1118,X1119,X1120,X1121,X1122,X1123,X1124,X1125,X1126,X1127,X1128,X1129,X1130,X1131,X1132,X1133,X1134,X1135,X1136,X1137,X1138,X1139,X1140,X1141,X1142,X1143,X1144,X1145,X1146,X1147,X1148,X1149,X1150,X1151,X1152,X1153,X1154,X1155,X1156,X1157,X1158,X1159,X1160,X1161,X1162,X1163,X1164,X1165,X1166,X1167,X1168,X1169,X1170,X1171,X1172,X1173,X1174,X1175,X1176,X1177,X1178,X1179,X1180,X1181,X1182,X1183,X1184,X1185,X1186,X1187,X1188,X1189,X1190,X1191,X1192,X1193,X1194,X1195,X1196,X1197,X1198,X1199,X1200,X1201,X1202,X1203,X1204,X1205,X1206,X1207,X1208,X1209,X1210,X1211,X1212,X1213,X1214,X1215,X1216,X1217,X1218,X1219,X1220,X1221,X1222,X1223,X1224,X1225,X1226,X1227,X1228,X1229,X1230,X1231,X1232,X1233,X1234,X1235,X1236,X1237,X1238,X1239,X1240,X1241,X1242,X1243,X1244,X1245,X1246,X1247,X1248,X1249,X1250,X1251,X1252,X1253,X1254,X1255,X1256,X1257,X1258,X1259,X1260,X1261,X1262,X1263,X1264,X1265,X1266,X1267,X1268,X1269,X1270,X1271,X1272,X1273,X1274,X1275,X1276,X1277,X1278,X1279,X1280,X1281,X1282,X1283,X1284,X1285,X1286,X1287,X1288,X1289,X1290,X1291,X1292,X1293,X1294,X1295,X1296,X1297,X1298,X1299,X1300,X1301,X1302,X1303,X1304,X1305,X1306,X1307,X1308,X1309,X1310,X1311,X1312,X1313,X1314,X1315,X1316,X1317,X1318,X1319,X1320,X1321,X1322,X1323,X1324,X1325,X1326,X1327,X1328,X1329,X1330,X1331,X1332,X1333,X1334,X1335,X1336,X1337,X1338,X1339,X1340,X1341,X1342,X1343,X1344,X1345,X1346,X1347,X1348,X1349,X1350,X1351,X1352,X1353,X1354,X1355,X1356,X1357,X1358,X1359,X1360,X1361,X1362,X1363,X1364,X1365,X1366,X1367,X1368,X1369,X1370,X1371,X1372,X1373,X1374,X1375,X1376,X1377,X1378,X1379,X1380,X1381,X1382,X1383,X1384,X1385,X1386,X1387,X1388,X1389,X1390,X1391,X1392,X1393,X1394,X1395,X1396,X1397,X1398,X1399,X1400,X1401,X1402,X1403,X1404,X1405,X1406,X1407,X1408,X1409,X1410,X1411,X1412,X1413,X1414,X1415,X1416,X1417,X1418,X1419,X1420,X1421,X1422,X1423,X1424,X1425,X1426,X1427,X1428,X1429,X1430,X1431,X1432,X1433,X1434,X1435,X1436,X1437,X1438,X1439,X1440,X1441,X1442,X1443,X1444,X1445,X1446,X1447,X1448,X1449,X1450,X1451,X1452,X1453,X1454,X1455,X1456,X1457,X1458,X1459,X1460,X1461,X1462,X1463,X1464,X1465,X1466,X1467,X1468,X1469,X1470,X1471,X1472,X1473,X1474,X1475,X1476,X1477,X1478,X1479,X1480,X1481,X1482,X1483,X1484,X1485,X1486,X1487,X1488,X1489,X1490,X1491,X1492,X1493,X1494,X1495,X1496,X1497,X1498,X1499,X1500,X1501,X1502,X1503,X1504,X1505,X1506,X1507,X1508,X1509,X1510,X1511,X1512,X1513,X1514,X1515,X1516,X1517,X1518,X1519,X1520,X1521,X1522,X1523,X1524,X1525,X1526,X1527,X1528,X1529,X1530,X1531,X1532,X1533,X1534,X1535,X1536,X1537,X1538,X1539,X1540,X1541,X1542,X1543,X1544,X1545,X1546,X1547,X1548,X1549,X1550,X1551,X1552,X1553,X1554,X1555,X1556,X1557,X1558,X1559,X1560,X1561,X1562,X1563,X1564,X1565,X1566,X1567,X1568,X1569,X1570,X1571,X1572,X1573,X1574,X1575,X1576,X1577,X1578,X1579,X1580,X1581,X1582,X1583,X1584,X1585,X1586,X1587,X1588,X1589,X1590,X1591,X1592,X1593,X1594,X1595,X1596,X1597,X1598,X1599,X1600,X1601,X1602,X1603,X1604,X1605,X1606,X1607,X1608,X1609,X1610,X1611,X1612,X1613,X1614,X1615,X1616,X1617,X1618,X1619,X1620,X1621,X1622,X1623,X1624,X1625,X1626,X1627,X1628,X1629,X1630,X1631,X1632,X1633,X1634,X1635,X1636,X1637,X1638,X1639,X1640,X1641,X1642,X1643,X1644,X1645,X1646,X1647,X1648,X1649,X1650,X1651,X1652,X1653,X1654,X1655,X1656,X1657,X1658,X1659,X1660,X1661,X1662,X1663,X1664,X1665,X1666,X1667,X1668,X1669,X1670,X1671,X1672,X1673,X1674,X1675,X1676,X1677,X1678,X1679,X1680,X1681,X1682,X1683,X1684,X1685,X1686,X1687,X1688,X1689,X1690,X1691,X1692,X1693,X1694,X1695,X1696,X1697,X1698,X1699,X1700,X1701,X1702,X1703,X1704,X1705,X1706,X1707,X1708,X1709,X1710,X1711,X1712,X1713,X1714,X1715,X1716,X1717,X1718,X1719,X1720,X1721,X1722,X1723,X1724,X1725,X1726,X1727,X1728,X1729,X1730,X1731,X1732,X1733,X1734,X1735,X1736,X1737,X1738,X1739,X1740,X1741,X1742,X1743,X1744,X1745,X1746,X1747,X1748,X1749,X1750,X1751,X1752,X1753,X1754,X1755,X1756,X1757,X1758,X1759,X1760,X1761,X1762,X1763,X1764,X1765,X1766,X1767,X1768,X1769,X1770,X1771,X1772,X1773,X1774,X1775,X1776,X1777,X1778,X1779,X1780,X1781,X1782,X1783,X1784,X1785,X1786,X1787,X1788,X1789,X1790,X1791,X1792,X1793,X1794,X1795,X1796,X1797,X1798,X1799,X1800,X1801,X1802,X1803,X1804,X1805,X1806,X1807,X1808,X1809,X1810,X1811,X1812,X1813,X1814,X1815,X1816,X1817,X1818,X1819,X1820,X1821,X1822,X1823,X1824,X1825,X1826,X1827,X1828,X1829,X1830,X1831,X1832,X1833,X1834,X1835,X1836,X1837,X1838,X1839,X1840,X1841,X1842,X1843,X1844,X1845,X1846,X1847,X1848,X1849,X1850,X1851,X1852,X1853,X1854,X1855,X1856,X1857,X1858,X1859,X1860,X1861,X1862,X1863,X1864,X1865,X1866,X1867,X1868,X1869,X1870,X1871,X1872,X1873,X1874,X1875,X1876,X1877,X1878,X1879,X1880,X1881,X1882,X1883,X1884,X1885,X1886,X1887,X1888,X1889,X1890,X1891,X1892,X1893,X1894,X1895,X1896,X1897,X1898,X1899,X1900,X1901,X1902,X1903,X1904,X1905,X1906,X1907,X1908,X1909,X1910,X1911,X1912,X1913,X1914,X1915,X1916,X1917,X1918,X1919,X1920,X1921,X1922,X1923,X1924,X1925,X1926,X1927,X1928,X1929,X1930,X1931,X1932,X1933,X1934,X1935,X1936,X1937,X1938,X1939,X1940,X1941,X1942,X1943,X1944,X1945,X1946,X1947,X1948,X1949,X1950,X1951,X1952,X1953,X1954,X1955,X1956,X1957,X1958,X1959,X1960,X1961,X1962,X1963,X1964,X1965,X1966,X1967,X1968,X1969,X1970,X1971,X1972,X1973,X1974,X1975,X1976,X1977,X1978,X1979,X1980,X1981,X1982,X1983,X1984,X1985,X1986,X1987,X1988,X1989,X1990,X1991,X1992,X1993,X1994,X1995,X1996,X1997,X1998,X1999,X2000,X2001,X2002,X2003,X2004,X2005,X2006,X2007,X2008,X2009,X2010,X2011,X2012,X2013,X2014,X2015,X2016,X2017,X2018,X2019,X2020,X2021,X2022,X2023,X2024,X2025,X2026,X2027,X2028,X2029,X2030,X2031,X2032,X2033,X2034,X2035,X2036,X2037,X2038,X2039,X2040,X2041,X2042,X2043,X2044,X2045,X2046,X2047,X2048,X2049,X2050,X2051,X2052,X2053,X2054,X2055,X2056,X2057,X2058,X2059,X2060,X2061,X2062,X2063,X2064,X2065,X2066,X2067,X2068,X2069,X2070,X2071,X2072,X2073,X2074,X2075,X2076,X2077,X2078,X2079,X2080,X2081,X2082,X2083,X2084,X2085,X2086,X2087,X2088,X2089,X2090,X2091,X2092,X2093,X2094,X2095,X2096,X2097,X2098,X2099,X2100,X2101,X2102,X2103,X2104,X2105,X2106,X2107,X2108,X2109,X2110,X2111,X2112,X2113,X2114,X2115,X2116,X2117,X2118,X2119,X2120,X2121,X2122,X2123,X2124,X2125,X2126,X2127,X2128,X2129,X2130,X2131,X2132,X2133,X2134,X2135,X2136,X2137,X2138,X2139,X2140,X2141,X2142,X2143,X2144,X2145,X2146,X2147,X2148,X2149,X2150,X2151,X2152,X2153,X2154,X2155,X2156,X2157,X2158,X2159,X2160,X2161,X2162,X2163,X2164,X2165,X2166,X2167,X2168,X2169,X2170,X2171,X2172,X2173,X2174,X2175,X2176,X2177,X2178,X2179,X2180,X2181,X2182,X2183,X2184,X2185,X2186,X2187,X2188,X2189,X2190,X2191,X2192,X2193,X2194,X2195,X2196,X2197,X2198,X2199,X2200,X2201,X2202,X2203,X2204,X2205,X2206,X2207,X2208,X2209,X2210,X2211,X2212,X2213,X2214,X2215,X2216,X2217,X2218,X2219,X2220,X2221,X2222,X2223,X2224,X2225,X2226,X2227,X2228,X2229,X2230,X2231,X2232,X2233,X2234,X2235,X2236,X2237,X2238,X2239,X2240,X2241,X2242,X2243,X2244,X2245,X2246,X2247,X2248,X2249,X2250,X2251,X2252,X2253,X2254,X2255,X2256,X2257,X2258,X2259,X2260,X2261,X2262,X2263,X2264,X2265,X2266,X2267,X2268,X2269,X2270,X2271,X2272,X2273,X2274,X2275,X2276,X2277,X2278,X2279,X2280,X2281,X2282,X2283,X2284,X2285,X2286,X2287,X2288,X2289,X2290,X2291,X2292,X2293,X2294,X2295,X2296,X2297,X2298,X2299,X2300,X2301,X2302,X2303,X2304,X2305,X2306,X2307,X2308,X2309,X2310,X2311,X2312,X2313,X2314,X2315,X2316,X2317,X2318,X2319,X2320,X2321,X2322,X2323,X2324,X2325,X2326,X2327,X2328,X2329,X2330,X2331,X2332,X2333,X2334,X2335,X2336,X2337,X2338,X2339,X2340,X2341,X2342,X2343,X2344,X2345,X2346,X2347,X2348,X2349,X2350,X2351,X2352,X2353,X2354,X2355,X2356,X2357,X2358,X2359,X2360,X2361,X2362,X2363,X2364,X2365,X2366,X2367,X2368,X2369,X2370,X2371,X2372,X2373,X2374,X2375,X2376,X2377,X2378,X2379,X2380,X2381,X2382,X2383,X2384,X2385,X2386,X2387,X2388,X2389,X2390,X2391,X2392,X2393,X2394,X2395,X2396,X2397,X2398,X2399,X2400,X2401,X2402,X2403,X2404,X2405,X2406,X2407,X2408,X2409,X2410,X2411,X2412,X2413,X2414,X2415,X2416,X2417,X2418,X2419,X2420,X2421,X2422,X2423,X2424,X2425,X2426,X2427,X2428,X2429,X2430,X2431,X2432,X2433,X2434,X2435,X2436,X2437,X2438,X2439,X2440,X2441,X2442,X2443,X2444,X2445,X2446,X2447,X2448,X2449,X2450,X2451,X2452,X2453,X2454,X2455,X2456,X2457,X2458,X2459,X2460,X2461,X2462,X2463,X2464,X2465,X2466,X2467,X2468,X2469,X2470,X2471,X2472	

COVER DESIGN BY ALDO GIORGINI